



Introduction à l'Infographie - Application à la visualisation des plantes

Frédéric Blaise

► To cite this version:

Frédéric Blaise. Introduction à l'Infographie - Application à la visualisation des plantes. 2004. inria-00107774

HAL Id: inria-00107774

<https://inria.hal.science/inria-00107774>

Submitted on 19 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Introduction à l'infographie

Application à la visualisation des plantes

Frédéric Blaise



(Cirad- Inria)



Mai 2004

Table des matières

1	INTRODUCTION	5
1.1	CREATION DE L'ATELIER DE MODELISATION DE L'ARCHITECTURE DES PLANTES (AMAP)	5
1.2	PRESENTATION GENERALE	6
2	MODELISATION, SIMULATION ET INFOGRAPHIE : UN PEU D'HISTOIRE	7
2.1	L'AVENEMENT DE L'INFORMATIQUE	7
2.1.1	EVOLUTION DE LA THEORIE	7
2.1.2	LES PREMIERES MACHINES A CALCULER	8
2.1.3	L'ERE DE L'ELECTRONIQUE : 5 GENERATIONS D'ORDINATEUR	9
2.2	MODELISATION ET SIMULATION	11
2.2.1	MODELISATION ET SIMULATION DANS LE VOCABULAIRE SCIENTIFIQUE	11
2.2.2	LA SIMULATION INFORMATIQUE	11
2.3	SIMULATION ET INFOGRAPHIE	12
3	LA VISION D'UNE SCENE VIRTUELLE PAR L'IMAGE DE SYNTHESE	15
3.1	MODELISATION DES OBJETS CONSTITUANT UNE SCENE	15
3.2	DETERMINATION DES SURFACES VISIBLES	16
3.3	MODELES D'ILLUMINATION ET COULEURS DES OBJETS	17
3.3.1	DESCRIPTION DES SOURCES LUMINEUSES	17
3.3.2	MODELES DE COULEUR	18
3.3.3	MODELES D'ILLUMINATION ET PROPRIETES DES MATERIAUX	19
3.3.4	INTERPOLATION DES COULEURS	20
3.4	APPLICATION DE TEXTURES	20
3.5	OMBRES PORTEES	21
3.6	TRANSPARENCE ET REFLEXION	22
3.7	DES MODELES D'ILLUMINATION GLOBALE	23
3.8	AFFICHAGE D'UNE SCENE 3D	24
4	METHODES INFORMATIQUES POUR LA SIMULATION ET LA REPRESENTATION DES PLANTES VIRTUELLES	27
4.1	DES MODELES INFORMATIQUES POUR LA REPRESENTATION DES PLANTES	27
4.1.1	DES MODELES DE PLANTES FIGURATIVES	27
4.1.2	LES L_ SYSTEMES	31
4.1.2.1	La théorie	31
4.1.2.2	Quelques applications des L-systèmes par P. Prusinkiewickz	32
4.2	DES MODELES BASES SUR LA BOTANIQUE POUR L'AGRONOMIE	34
4.2.1	MODELES AGRONOMIQUES ET PLANTES VIRTUELLES	35
4.2.1.1	Les différents modèles utilisés dans ce domaine	35
4.2.1.2	Exploitation des maquettes tridimensionnelles	35
4.2.2	LES LOGICIELS DE SIMULATIONS AMAP	35
4.2.2.1	Concepts de base	35
4.2.2.2	Données nécessaires à la simulation	36
4.2.2.3	Simulation de la croissance	36

4.2.2.4	Différentes générations de logiciel	36
4.2.2.5	Des structures informatiques adaptées	37
4.2.3	TOPOLOGIE ET GEOMETRIE	38
4.2.4	DYNAMIQUE DE CROISSANCE	39
4.2.4.1	La simulation à événements discrets : aspect théorique	40
4.2.4.2	Application de la simulation à événements discrets à la problématique plante	41
4.2.4.3	Une structure informatique pour l'échéancier	42
4.2.4.4	Mise à jour de l'échéancier	44
4.2.5	INTERACTIONS PLANTE/ENVIRONNEMENT	44
4.2.5.1	Discrétisation de l'espace et des plantes	45
4.2.5.2	La plante dans un espace voxel	46
4.2.5.3	Méthode de discrétisation des entre-nœuds	46
4.2.5.4	Traitement de la compétition pour l'espace	48
4.2.5.5	Mise en place d'obstacles dans l'espace voxel	50
4.2.5.6	Discrétisation de la voûte céleste	51
4.2.5.7	Influence de la lumière	52
4.2.5.8	Conclusion	53

5 ARCHITECTURE DES PLANTES ET PAYSAGISME **55**

5.1	DES OUTILS DE CAO POUR LA CONSTRUCTION DES PLANTES	55
5.1.1	LE LOGICIEL TREE PROFESSIONAL (<i>ONYX COMPUTING</i>)	55
5.1.1.1	Concepts généraux	55
5.1.1.2	Des outils spécialisés	56
5.1.1.3	Quelques réalisations	57
5.1.2	LE LOGICIEL XFROG (<i>GREENWORKS</i>)	58
5.1.2.1	Concepts généraux	58
5.1.2.2	Quelques réalisations	59
5.2	DES OUTILS DE MODELISATION POUR LE PAYSAGE	60
5.2.1	AMAP GENESIS : UN MODELEUR BOTANIQUE	61
5.2.1.1	Principe de fonctionnement	61
5.2.1.2	Structuration des données botaniques	62
5.2.1.3	Création d'une plante virtuelle	65
5.2.1.4	Fonctionnalités du logiciel Genesis	65
5.2.1.5	Le catalogue des plantes virtuelles	68
5.2.2	CONCEPTION ET VISUALISATION D'UN PAYSAGE VIRTUEL	72
5.2.2.1	Création d'un terrain : AMAP Altis	72
5.2.2.2	Conception d'un paysage : AMAP Orchestra	73
5.2.2.3	Quelques réalisations de projets d'aménagement	76

6 CONCLUSION **79**

7 BIBLIOGRAPHIE **81**

1 INTRODUCTION

1.1 CREATION DE L'ATELIER DE MODELISATION DE L'ARCHITECTURE DES PLANTES (AMAP)

A ses débuts, juste après l'apparition des premiers ordinateurs modernes (Cf.2.1), l'infographie se limitait à l'affichage de données sur des tables traçantes ou sur des écrans à tube cathodique. Jusqu'au début des années 80, son champ d'application était restreint et très spécialisé, en grande partie à cause du coût du matériel et des logiciels, d'ailleurs très peu nombreux, que proposait le marché de l'infographie. L'arrivée des micro-ordinateurs personnels équipés d'écran *raster* (l'écran est un ensemble de points ou pixels que l'on allume ou pas avec une certaine couleur) favorisa le développement des *bitmap graphics* (mémoire dite graphique de l'ordinateur contenant, pour chacun des pixels de l'écran, sa valeur). Ce développement matériel banalisa l'utilisation de l'infographie et permit la création de logiciels graphiques performants et très interactifs : tableurs, traitements de texte, logiciels de dessin ...

Ainsi, les domaines d'application de l'infographie ne cessèrent de se multiplier, incluant maintenant la création, le stockage et la manipulation de modèles et d'images d'objets. Lorsque l'on parle de création d'image en infographie, on utilise le terme d'image de synthèse. On peut dire que l'image de synthèse est à l'infographie, ce que l'hyperréalisme est à la peinture : on cherche à représenter un objet de telle façon qu'il soit très difficile de différencier le vrai du faux. Ceci implique l'utilisation, en image de synthèse, de techniques spécifiques et très performantes (Cf.1). Ces dernières peuvent être regroupées en deux catégories principales :

1. la modélisation : on cherche ici à représenter l'objet considéré sous forme de données informatiques (Cf.3.1)
2. le rendu : on regroupe sous ce terme toutes les techniques infographiques nécessaires à rendre l'aspect de l'objet en termes d'apparence, de couleur et de texture (Cf.3.2 à 3.6)

La modélisation est une étape importante, voire primordiale, lors de la conception d'une image de synthèse. Plus le modèle d'un objet est fin, moins le rendu prend d'importance dans la qualité de l'image obtenue, bien qu'un minimum de techniques soit cependant indispensable. Si les modélisateurs 3D classiquement utilisés en CAO sont suffisants pour rendre compte d'objets manufacturés (voiture, avion, bâtiment ...), il n'en va pas de même pour la représentation des phénomènes naturels qui échappent aux possibilités de tels modélisateurs. Ces objets ou phénomènes naturels sont pourtant indispensables si l'on veut représenter des scènes qui comprennent autre chose que des objets, des machines, des bâtiments ou des terrains. S'appuyant sur le principe que toute la nature pouvait être représentée par des algorithmes simples capables de générer les formes les plus complexes, les informaticiens se sont attaqués sans complexe à toutes sortes de défis comme la modélisation des nuages, des vagues, du feu, de la pluie, etc.

Les plantes n'ont pas échappé à cette investigation et ont été passées au crible de techniques nombreuses et diverses (Cf.4.1). Toutefois, les résultats obtenus se sont avérés peu convaincants pour un botaniste. La raison en est simple : malgré un rendu souvent de grande qualité, les images obtenues laissent apparaître la pauvreté des modèles utilisés. Seule la prise en compte des lois de la morphogenèse végétale lors de la conception du modèle assure une bonne adéquation avec la réalité botanique.

Les informaticiens Jean Françon¹ et Claude Puech², spécialistes en images de synthèse, se sont alors intéressés aux travaux de l'agronome Philippe de Reffye³ sur la simulation de l'architecture des plantes

¹ Professeur au Département d'Informatique de l'Université Pasteur à Strasbourg

² Directeur de l'unité de recherche Futurs à l'INRIA

appliquée à une problématique agronomique. Ces travaux, utilisant la notion d'architecture des plantes définie par les botanistes Francis Hallé⁴ et R. Oldeman⁵ pour décrire la structure morphogénétique des végétaux, leur sont apparus comme parfaitement adaptés à la modélisation des végétaux, en particulier pour l'image de synthèse. C'est cette collaboration interdisciplinaire qui est à l'origine de la création de l'Atelier de Modélisation de l'Architecture des Plantes (AMAP) au sein du CIRAD à Montpellier en 1984.

En relation avec la communauté scientifique nationale et internationale, l'UMR **AMAP** « botanique et bioinformatique de l'Architecture des Plantes » (UMR 5120 du Cnrs, UMR 931 de l'INRA, R123 de l'IRD) associe des agents de plusieurs organismes (Cirad-Cnrs-Ephe-Inra-Inria-Ird-Université de Montpellier II) dont certains sont affectés outremer ou à l'étranger. L'UMR conçoit, développe et évalue des modèles d'analyse et de suivi de l'évolution des paramètres morphologiques, anatomiques et fonctionnels du développement architectural d'espèces végétales isolées ou en peuplement. L'UMR est organisée en 3 équipes couplées par un champ méthodologique transversal et ses recherches mettent en jeu des disciplines variées -- biologie, botanique, écologie, mathématiques appliquées, informatique --, dont l'association constitue un enjeu scientifique et technique majeur pour l'agronomie et la foresterie modernes. Les méthodes et les outils que l'UMR élabore sont mis à la disposition des chercheurs et des étudiants qu'il accueille au sein de ses équipes. Ils sont aussi diffusés sous forme d'enseignements ou de logiciels dont certains sont commercialisés (Amap, 2003).

1.2 PRESENTATION GENERALE

En consultant le site Web de l'UMR Amap (Amap, 2003), on peut voir qu'il est possible d'aborder l'analyse d'une structure aussi complexe que peut l'être une plante, à condition toutefois de disposer de connaissances spécifiques suffisantes (analyse architecturale) et d'un cadre théorique adapté (codage de la topologie et analyse mathématique : AMAPmod).

« Les formes dans la nature ne seront pas définies en mesurant des données 3D de certaines prises comme modèles, mais en expliquant mathématiquement des règles variées de formation, et en recréant des modèles fondés sur ces théories. »

Yoichiro Kawaguchi (Robertson, 1989)

La mise en œuvre de modèles informatiques capables de rendre compte de l'architecture d'une plante, voire de simuler la dynamique de sa croissance, se heurte aux mêmes difficultés et nécessite donc la définition d'algorithmes qui s'appuient sur les résultats issus des analyses botanique et mathématique.

Toutefois, après avoir brièvement évoqué l'avènement de la simulation informatique, puis passé en revue les différentes approches utilisées en informatique pour modéliser les plantes, nous verrons dans les paragraphes suivants que la construction des modèles informatiques dédiés aux plantes dépend fortement des objectifs visés :

- représentation picturale de la plante pour des applications purement graphiques,
- utilisation de plantes virtuelles en biologie. Dans ce cas, nous distinguerons différentes catégories en fonction des particularités du domaine d'application (agronomie, sylviculture, télédétection, aménagement paysager ...).

³ Docteur d'Etat, agronome au Centre de Coopération International en Recherche pour le Développement (CIRAD) à Montpellier

⁴ Professeur de botanique à l'Institut de Botanique de Montpellier

⁵ Professeur de sylviculture à l'université de Wageningen

2 MODELISATION, SIMULATION ET INFOGRAPHIE : UN PEU D'HISTOIRE

2.1 L'AVENEMENT DE L'INFORMATIQUE⁶

2.1.1 Evolution de la théorie

En 1700 Av. J.C., l'empereur de Mésopotamie Hammourabi introduisit la notion de normes dans le traitement de l'information. Quant au philosophe grec Aristote, il fut à l'origine de la logique mathématique (300 Av. J.C.).



Leibnitz
(1646/1716)

Il fallut attendre la fin du XVII^{ème} siècle pour voir le binaire faire son apparition en Europe. C'est Gottfried Wilhelm von Leibnitz qui, en 1697, présenta ses travaux sur la *Dyadique* devant l'Académie des Sciences de Paris (Leibnitz, 1697).



Aristote
(-384/-322)

En 1843, Ada Lovelace établit la théorie de la programmation et inventa le terme « algorithme », en l'honneur du mathématicien arabe Al Khwarizmi, pour définir un processus logique permettant l'exécution d'un programme.



George Boole
(1815/1864)

En 1854, le mathématicien George Boole (Boole, 1854) définit la théorie de la logique binaire : il expliqua qu'un système à deux états (vrai et faux, zéro et un ...) suffit à décrire un raisonnement.



Ada Lovelace
(1814/1852)

C'est à Torres Y Quevedo que l'on doit la définition théorique de l'automate (1914). Mais c'est Alan Turing qui, en 1936, fut à l'origine d'une étape importante dans l'histoire de l'informatique : il définit la notion d'algorithme et décrit une machine virtuelle (la machine de Turing) capable de résoudre tout problème énoncé par un algorithme.



George Stibitz
(1904/1995)

Mis au point conjointement par George Stibitz aux Etats-Unis et Konrad Zuse en Allemagne (1940), le système de calcul en virgule flottante permit d'exprimer les grands nombres sous forme d'une mantisse et d'un exposant et, par conséquent, d'économiser de la mémoire.



Alan Turing
(1912/1954)



Konrad Zuse
(1910/1995)

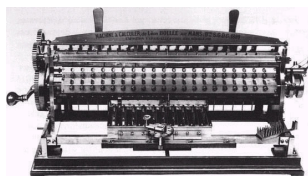
En 1948, Claude Elwood Shannon applique dans ses travaux de thèse (Shannon, 1948) la théorie de Boole. Il définit le terme de bit (Binary digIT) pour mesurer la quantité d'information.



Claude Shannon
(1916/2001)

⁶ D'après François Guillier, 1998.

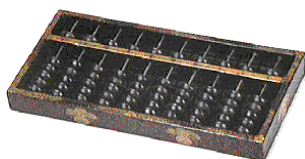
2.1.2 Les premières machines à calculer



Machine de Bollée (1887)

Des boules et des jetons en argile inventés par les Sumériens (10000 Av. J.C.) jusqu'à la machine à multiplier de Léon Bollée (1887), l'histoire du traitement de l'information et de l'automatisation est ponctuée d'inventions qui ont permis à l'homme d'augmenter ses possibilités de calcul et d'aller plus loin dans le domaine de la recherche.

Ainsi, au boulier chinois (3000 Av. J.C.) et à l'abaque romain (version occidentale du



Boulier chinois (-3000)

boulier) (100 Apr. J.C.), a succédé la machine à calculer de Wilhelm Schickard (1623) capable de réaliser des additions et des soustractions. Elle était basée sur le principe des bâtons de Néper (1615) qui étaient composés de 10 cases divisées par une diagonale; dans le triangle inférieur droit figurait le chiffre des unités du produit et dans le supérieur gauche, le chiffre des dizaines.



Jean Néper
(1550/1617)



Blaise Pascal
(1623/1662)

La première machine à calculer qui fut vraiment utilisée a été mise au point par Blaise Pascal en 1641 : la Pascaline. Son principe fut repris et amélioré par Leibnitz en 1694.



La Pascaline (1641)

En 1728, Falcon inventa les premiers métier à tisser qui utilisaient un *programme* sur plaquettes de bois perforées. C'est à partir de cette invention que Joseph-Marie Jacquard réalisa des métiers à tisser automatisés commandés par des cartes perforées qui équipèrent les ateliers lyonnais (1806).



Charles Babbage
(1791/1871)

En 1820, Charles Babbage construisit un calculateur (la machine à différences) utilisant le système décimal : la manivelle faisait tourner les roues par dixièmes de tour. En 1833, avec l'aide de Lady Lovelace, Babbage conçut la machine analytique et définit les différents éléments qui, pour lui, devait entrer dans la composition d'une machine à calculer :

- un dispositif permettant d'introduire les données numériques,
- une mémoire pour conserver les valeurs numériques entrées,
 - une unité de commande afin que l'utilisateur puisse diriger la machine,
 - un « *moulin* » chargé d'effectuer les calculs,
 - un dispositif permettant de prendre connaissance des résultats.



Machine à différences
(1820)

Un siècle plus tard, les premiers ordinateurs seront construits sur ces mêmes principes, ce qui permet à Babbage d'être considéré comme le père des ordinateurs modernes.



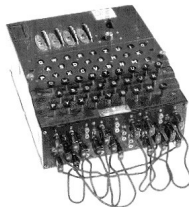
Herman Hollerith
(1860/1929)

En 1884, Herman Hollerith met au point une machine à statistiques à cartes perforées pour effectuer le recensement américain de 1890. Cette machine sera à l'origine de la fondation de la société « *Tabulating Machine Company* » (1896) qui deviendra plus tard « *International Business Machines* » (IBM).

2.1.3 L'ère de l'électronique : 5 générations d'ordinateur

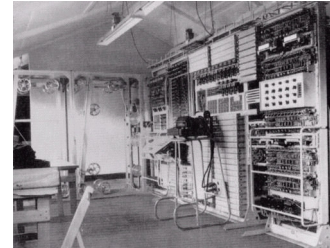
On subdivise d'ordinaire cette période de l'histoire des ordinateurs en fonction des découvertes techniques qui en ont permis l'amélioration. Il en résulte 5 générations d'ordinateur.

La première génération, de 1945 à 1956, se caractérisait par leur spécialisation (les machines étaient conçues pour effectuer une tâche bien précise) et par la spécificité de leur langage machine. C'était aussi le temps des tubes à vide et des tambours magnétiques, ce qui expliquait leur grande taille. Cette génération doit son développement à la seconde guerre mondiale.



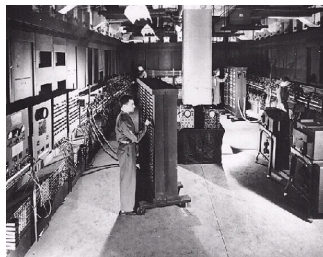
L'Enigma (1930)

En effet, Alan Turing participa à la réalisation de la *Bombe*, première machine électromécanique de décryptage des messages codés avec l'*Enigma* allemande, puis à celle du Colossus I, en 1943.

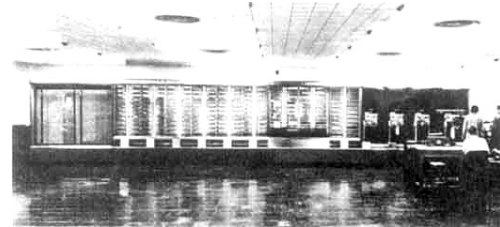


Colossus I (1943)

La même année, à partir des travaux de Babbage, Howard H. Aiken conçut le Harvard MARK 1, premier calculateur électromécanique à registre, qui permettait à l'U.S. Navy de disposer de tableaux de correspondances balistiques. Un autre ordinateur dont le développement avait été accéléré pendant la guerre était l'ENIAC (*Electronic Numerical Integrator and Computer*). Développé par John Presper Eckert et John W. Mauchly, il avait été commandé par l'armée des États-Unis en 1943 pour effectuer les calculs de balistique.

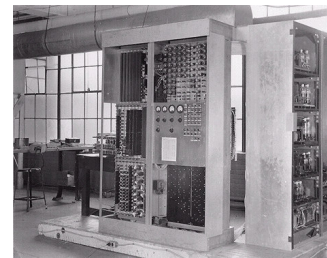


ENIAC (1943)



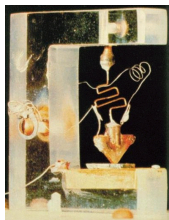
Harvard MARK 1 (1943)

En 1945, von Neumann construisit l'EDVAC (*Electronic Discrete Variable Automatic Computer*), premier calculateur à programme enregistré. Enfin, en 1951, l'UNIVAC (*UNIVersal Automatic Computer*) devint un des premiers ordinateurs à être commercialisés.

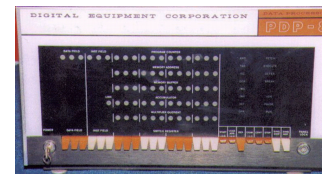


EDVAC (1945)

En 1948, l'invention du transistor annonça la venue de la seconde génération (de 1956 à 1963). Le remplacement des tubes à vides par les transistors modifia le développement des ordinateurs, leur taille notamment. Si le premier ordinateur transistorisé fut le Tradic, en 1955, le premier mini-ordinateur diffusé massivement fut le PDP-8 (*Programmed Data Processor*) de DEC en 1965. Des langages de haut niveau firent également leur apparition durant cette période (COBOL, FORTRAN) et permirent le développement de l'industrie du logiciel.



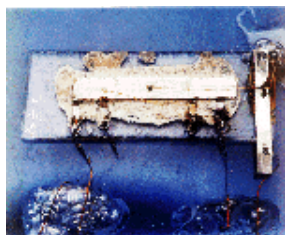
Transistor à pointe (1948)



PDP-8 (1965)

La troisième génération (de 1964 à 1971) est marquée par l'utilisation du Quartz et le développement des circuits intégrés, en remplacement des transistors. Apparaissent également les systèmes d'exploitation (*Operating System*) qui permettent de faire tourner plusieurs programmes différents sur une même machine avec un programme central contrôlant et coordonnant la mémoire centrale de l'ordinateur.

Avec l'apparition des circuits intégrés, la quatrième génération (de 1971 à nos jours) se caractérise par une réduction toujours plus grande de ces circuits. Cette réduction des circuits permet la création d'ordinateurs



*Le premier circuit intégré
(Jack Kilby, 1958)*

toujours plus petits et toujours moins chers. Ainsi, en 1973, apparaît le premier micro-ordinateur du monde, le Micral-N, inventé par André Truong pour la société française R2E. Mais les Américains préfèrent considérer l'Altair 8800 de Roberts, créé en 1975, comme le véritable premier micro-ordinateur. Il est vrai que ce dernier bénéficia du premier BASIC Microsoft. Cette génération donna naissance à de nombreux micro-ordinateurs dont les plus célèbres sont l'Apple II de Steve Jobs et Stephen Wozniack en 1977, le Sinclair ZX81 en 1977, l'IBM-PC en 1981, le Lisa d'APPLE en 1983, l'Amiga (spécialisé dans le

graphisme et la musique) en 1984, le Macintosh d'APPLE en 1984 et le PowerMac



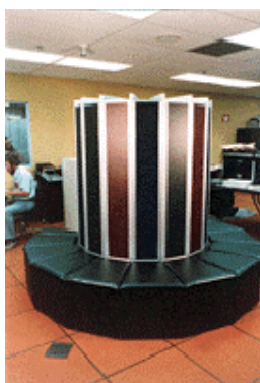
IBM-PC (1981)

d'APPLE en 1994. A côté de ces micro-ordinateurs furent développés les premiers ordinateurs pour le calcul intensif : machines vectorielles, machines massivement parallèles ... La famille la plus connue est sans nul doute la série des CRAY : CRAY I de Seymour Cray en 1976, CRAY X-MP en 1982, CRAY II en 1985. D'autres machines spécialisées ont fait leur apparition, notamment dans le domaine du graphisme et de l'image de synthèse. La famille des Silicon Graphics, qui a d'ailleurs repris en 1994 le développement des machines CRAY pour le calcul



Apple II (1977)

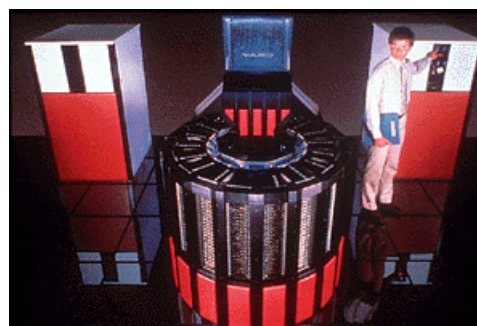
intensif, est un bon exemple de machines de ce type.



CRAY I (1976)



CRAY-X MP (1982)



CRAY II (1985)

On aura compris que les ordinateurs de la cinquième génération sont en cours de développement. Il est donc difficile de les décrire exhaustivement, mais il est possible d'esquisser certains traits qui devraient les caractériser : utilisation de la technologie des supraconducteurs pour améliorer les performances (de transmission des données notamment), parallélisme dans les traitements, interface homme-machine graphique et vocale (en temps réel !), capacité d'apprentissage et de décision grâce à l'intelligence artificielle. On pourra remarquer que certaines de ces caractéristiques font déjà partie intégrante de machines existantes.

2.2 MODELISATION ET SIMULATION

2.2.1 Modélisation et simulation dans le vocabulaire scientifique

Avant d'aborder la simulation informatique, il est nécessaire d'expliciter les termes *modélisation* et *simulation*. En effet, trop souvent, ces deux mots sont à l'origine de confusion et d'incompréhension, notamment entre spécialistes de domaines différents. Ainsi, il n'est pas rare d'entendre des chercheurs s'appropriier les termes modélisation et simulation en les définissant suivant des critères propres à leur domaine de compétence. Cette vision partisane ne peut que compliquer encore un peu plus les échanges entre des disciplines certes différentes, mais qui doivent travailler de concert en de nombreuses occasions.

Bien que *typée* par le champ disciplinaire de son application, la modélisation consistera toujours à construire un modèle, à savoir *une structure formalisée utilisée pour rendre compte d'un ensemble de phénomènes qui possèdent entre eux certaines relations*. Cette définition claire est suffisamment générale pour s'adapter à tout domaine de recherche, indépendamment des particularismes.

La simulation est très liée, voire même inféodée, à la notion de modèle : c'est *une méthode de mesure et d'étude consistant à remplacer un phénomène, un système à étudier par un modèle plus simple, mais ayant un comportement semblable*. Par conséquent, si on peut dire qu'une simulation est efficace lorsqu'elle permet d'obtenir des résultats semblables aux caractéristiques du phénomène observé, on ne peut pas pour autant affirmer que les choix faits dans la simulation expliquent complètement le fonctionnement du phénomène.

Nous terminerons par cette définition philosophique de la notion de modèle :

« Ce que la métaphore est au langage poétique, le modèle l'est au langage scientifique. Métaphores et modèles sont des fictions que nous nous plaisons à construire soit pour la joie de l'étincelle, soit pour décrire plus adéquatement le réel. »

Philippe Quéau (Quéau, 1986)

2.2.2 La simulation informatique

L'avènement des ordinateurs très rapides a bouleversé les méthodes scientifiques. Traditionnellement, les chercheurs élaboraient des modèles idéaux dont ils s'efforçaient de tirer des prévisions vérifiables par l'expérimentation ou l'observation, mais cette méthode avait ses limites : il est en effet difficile de simplifier la description de phénomènes complexes, car certains éléments clés du phénomène résultent de la combinaison ou de l'amplification de causes minimes. En fait, si on simplifie trop un modèle, on risque de perdre de vue la nature même du phénomène.



Johannes von
Neumann

L'utilisation moderne du terme *simulation* trouve ses origines dans les travaux menés par Johannes von Neumann et Stanislaw Ulam à la fin des années 40, quand ils appliquèrent la méthode d'*analyse de Monte Carlo* à des problèmes ayant trait à la physique nucléaire, problèmes dont la solution était soit trop coûteuse à expérimenter, soit trop complexe à établir par un traitement analytique. L'analyse de Monte Carlo résout un problème mathématique non probabiliste, en simulant un processus stochastique dont les distributions de probabilité satisfont les relations mathématiques du problème non probabiliste.



Stanislaw Ulam

Avec la venue des ordinateurs puissants au début des années 50, la simulation a pris encore une nouvelle forme, car il était maintenant possible d'expérimenter des modèles mathématiques sur des calculateurs.

Pour la première fois, des statisticiens et des sociologues pouvaient, comme les physiciens, effectuer des expériences contrôlées comme en laboratoire. Ils utilisaient ainsi des calculateurs électroniques en lieu et place d'outils plus concrets, comme par exemple, des réacteurs nucléaires qu'auraient utilisés les physiciens.

L'avènement de la simulation informatique fit que de nombreuses applications prirent corps, et même, qu'un grand nombre de problèmes pratiques et théoriques firent leur apparition en même temps que cette innovation. Aujourd'hui, la puissance sans cesse croissante des ordinateurs permet au chercheur d'étudier des phénomènes complexes comme la diffusion des porteurs de charge électrique dans les semi-conducteurs ou la collision entre des galaxies comptant des millions d'étoiles. Avec les simulations, il est possible de prévoir les conséquences de théories simples dans des conditions très complexes. Par ailleurs, sans procéder à des simplifications ou à des approximations, la simulation peut permettre de mesurer des paramètres comme si l'expérience avait été effectivement réalisée. Ainsi, des expériences irréalisables peuvent être simulées et des paramètres incalculables par expérimentation peuvent être estimés.

2.3 SIMULATION ET INFOGRAPHIE

Plus récemment, au début des années 70, est apparue une nouvelle forme de modélisation qui débouche sur la création de représentations graphiques (image de synthèse) : *l'infographie*.



Figure 1 Rhinocéros affrontés de la Grotte Chauvet (France).

Si on considère la synthèse d'image sous sa seule forme figurative, elle participe au même titre que d'autres techniques graphiques à la représentation d'objets et de formes. Prise comme modèles, les images de synthèse entrent dans le domaine de la simulation, univers sinon distinct, du moins différent du monde réel. Elles semblent, de plus, être dotées d'une efficacité symbolique sans précédent. De fait, bien plus que les autres images (picturales, photochimiques ou vidéos), les images numériques sont en mesure de représenter des phénomènes concrets autant que d'autres beaucoup plus abstraits. L'intérêt de l'image numérique pour la simulation apparaît alors assez clairement. L'image de synthèse est un moyen puissant de rendre compte de résultats de simulations, concrets, palpables ou plus abstraits, sous des formes diverses, souvent remarquables par leur technique et toujours démonstratifs par leur présentation (Fig.2).

Les images que nous sommes habitués à manipuler, regarder, fabriquer, sont extrêmement variées quant à la manière dont elles sont fabriquées. De l'empreinte gravée à la trace laissée par le crayon, des trames superposées de l'imprimerie au grain de la photographie, il existe d'innombrables possibilités pour produire des images. Toutes ces images ont en commun d'être tributaires d'un support, même si ces supports sont d'une grande variété : papier, toile, pierre (Fig.1), murs, carton.

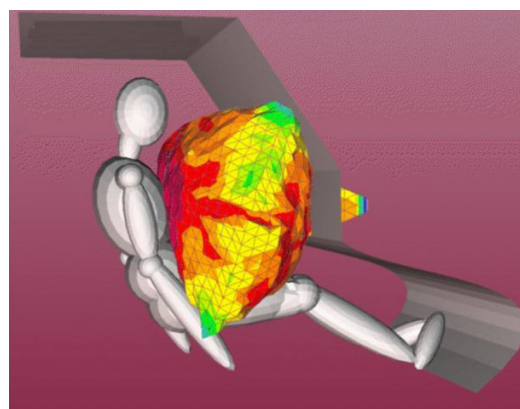


Figure 2 Simulation d'un choc mécanique sur un mannequin (MacNeal-Schwendler Corporation, Los Angeles, CA).

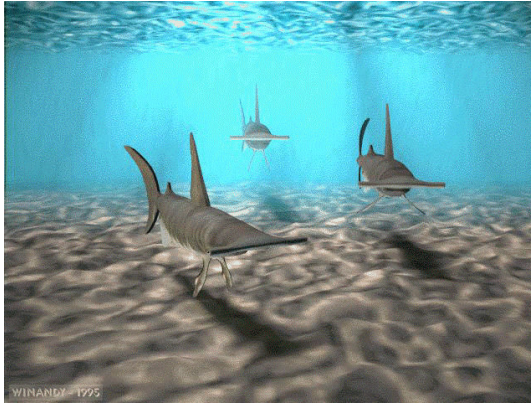


Figure 3 Modélisation d'un fond sous-marin
(*Ecosystem simulation, Université de Clermont-Ferrand*).

Au niveau de développement actuel de l'infographie, le besoin d'introduire un plus grand réalisme dans la génération des images se fait sentir. Ceci est particulièrement vrai dans le domaine de la représentation de phénomènes naturels (Fig.3). Leur représentation conduit à de nombreux problèmes dus à la nature même de ces objets. Ainsi, la connaissance superficielle de leurs propriétés physiques, de même que leur caractère dynamique n'en permet pas une modélisation aisée. Toutefois, de nombreux modèles informatiques ont été développés afin de simuler et de représenter les plantes, entités biologiques particulièrement complexes.

3 LA VISION D'UNE SCENE VIRTUELLE PAR L'IMAGE DE SYNTHÈSE

Nous avons évoqué, dans le paragraphe précédent (Cf.2.3), l'avènement des techniques graphiques dans le monde de l'informatique. Dans ce chapitre, nous allons aborder un aspect important de l'infographie : l'image de synthèse, autrement dit la création d'images réalistes de scène en trois dimensions (3D). On peut qualifier de *réaliste* une image qui intègre les interactions entre les sources lumineuses (que ce soit le soleil ou des sources *locales* telles des lampadaires ou des lampes de bureau) et les objets physiques constituant la scène.

La création d'images de synthèse est devenue un enjeu important dans de nombreux domaines : la simulation, les jeux, la recherche et l'éducation, la commande et le contrôle. Ainsi, les simulateurs de vol doivent non seulement générer des images de grande qualité (Fig.4), très proches de la réalité, mais aussi permettre un changement dynamique de la vision des paysages survolés par le pilote depuis son cockpit. De même, les dessinateurs industriels souhaitent pouvoir visualiser leurs créations (Fig.5) avant leur fabrication, afin d'augmenter l'efficacité des études menées et d'en minimiser les coûts. Quant à la recherche, elle utilise de plus en plus l'image de synthèse comme un outil d'analyse des phénomènes observés ou de vérification des théories élaborées (Fig.6).



Figure 4 Pilote dans un simulateur de vol (NASA).



Figure 5 Maquette virtuelle de la voiture de James Bond (CATIA-CADAM Dassault Systèmes).

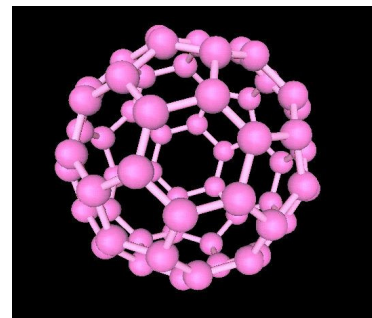


Figure 6 Visualisation d'une molécule de fullerene C60 (CNUSC).

La création d'images de synthèse nécessite un certain nombre d'étapes techniques que nous allons maintenant détailler (d'après Foley *et al.*, 1990).

3.1 MODELISATION DES OBJETS CONSTITUANT UNE SCENE

Indépendamment des techniques de rendu utilisées, le réalisme d'une image dépend fortement des choix effectués lors de la modélisation des différents objets constituant la scène.

Le modèle le plus communément utilisé est le modèle géométrique. Dans ce modèle, chaque face (encore appelée polygone) de l'objet est décrite : nombre de points la définissant, coordonnées de ces points, définition du vecteur normal en chaque point de chaque face (Fig.7). Si cette technique est simple d'emploi, elle n'est cependant pas très efficace dans le cas de phénomènes naturels (nuage, feu, eau, arbre ...) ou pour des objets à l'architecture complexe (pont de Normandie, Tour Eiffel ...). En effet, dans le

cas du feu par exemple, sa forme changeante à la géométrie complexe est difficilement modélisable à l'aide de polygones. De même, le nombre de poutres, de boulons, de rivets utilisés dans la construction de la Tour Eiffel rend sa description par un modèle géométrique trop fastidieuse.

Une autre approche a été définie afin de permettre la description de phénomènes complexes : on cherche à représenter un large éventail d'objets par un modèle unique aisément ajustable à l'aide de paramètres. Ces paramètres deviennent ainsi les données à partir desquelles le modèle est généré. Cette technique, appelée *database amplification* (Smith, 1984), permet de décrire à l'aide d'un modèle unique des phénomènes semblables à un haut niveau d'appréhension. C'est le cas par exemple du modèle AMAP qui s'applique à tout le règne végétal malgré les différences notables entre les différents groupes de plantes (herbacées, graminées, arbustes, arbres) et les différentes espèces d'une même famille (épicéa et sapin, peuplier et chêne ...). Mais une plante reste une entité biologique végétale dont la morphogenèse et le fonctionnement restent fondamentalement les mêmes d'un individu à un autre bien que la valeur des paramètres les décrivant varie.

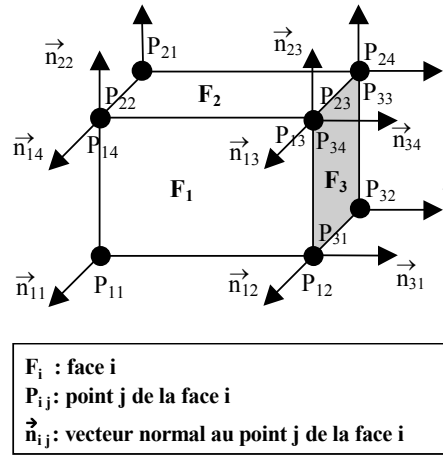


Figure 7 Modèle géométrique.

De nombreux modèles peuvent être qualifiés de *database amplification* :

- Les modèles procéduraux décrivent des objets qui peuvent être modifiés par des événements extérieurs. Magnenat-Thalman et Thalman (1985) décrivent un modèle procédural pour les ponts dans lequel un pont est constitué d'une route, d'une superstructure, de piliers et de parapets, chacun des éléments étant défini par des paramètres.
- Les modèles fractals (Cf. Fig.7,8) sont souvent utilisés pour représenter des objets naturels : des montagnes (Fournier *et al.*, 1982), des fougères (Cf. Fig.12).
- Les modèles basés sur des grammaires tels les L-systèmes (Cf.4.1.2) sont souvent utilisés pour générer des plantes.
- Les systèmes de particule ont été également utilisés pour la génération de plantes et de forêts (Cf. Fig.11).
- Les modèles basés sur la physique prennent en considération des lois de la physique (contraintes mécaniques, gravité, frottement ...) pour déterminer la forme d'objets tels les tissus ou des systèmes mécaniques soumis à des contraintes dynamiques.
- D'autres modèles utilisent les lois de la physique pour décrire des phénomènes naturels : les vagues (Fournier et Reeves, 1986), les nuages (Gardner, 1985), ... Mais, à la différence des modèles précédents, ils utilisent des lois physiques qui peuvent être très éloignées de la réalité du phénomène modélisé.

3.2 DETERMINATION DES SURFACES VISIBLES

Une fois les objets constituant la scène modélisés, il faut les afficher à l'écran. Une des premières tâches consiste à déterminer quelles sont les faces visibles, c'est à dire les faces qui ne sont pas cachées par la présence d'un autre objet ou d'une autre face du même objet entre l'observateur et la face considérée. Cette étape est primordiale pour la compréhension de l'image. Si l'on examine la figure 8, dans le cas de la non détermination des faces cachées, on peut se demander si la face en rouge est devant ou derrière la face en bleu, alors que, lorsque l'on élimine les faces cachées, le doute n'est plus permis.

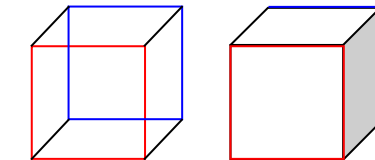


Figure 8 Vue d'un cube sans et avec élimination des faces cachées.

De nombreux algorithmes ont été développés afin de traiter ce problème : algorithmes de Warnock (1969), de Watkin (1970), de Galimberti et Montanari (1969), du *z-buffer* encore appelé *depth-buffer*... La technique de *z-buffer*, développé par Catmull (1974), nécessite de disposer d'une mémoire tampon (*buffer*) dans laquelle sera stockée, pour chaque pixel, la distance le séparant de l'observateur. Cette mémoire a donc une taille identique à celle utilisée pour l'affichage des pixels sur l'écran (*frame buffer*) et peut être cablée dans la machine (*hardware buffer*) ou implémentée par logiciel (*software buffer*). Son utilisation est très simple : à chaque fois que l'on doit afficher un point constituant un polygone, après calcul de sa projection sur l'écran (Cf.3.7), on vérifie que sa distance à l'observateur est inférieure à celle du pixel déjà affiché.

3.3 MODELES D'ILLUMINATION ET COULEURS DES OBJETS

Une fois les faces cachées éliminées, il faut calculer la couleur des faces retenues. Leur apparence dépend du type des sources lumineuses qui les éclairent, de leurs propriétés (couleur, texture, réflectance), de leur position et de leur orientation par rapport aux sources lumineuses, à l'observateur et aux autres surfaces de la scène. Ainsi, une surface qui ne reflète que la lumière bleue apparaîtra noire si elle est éclairée par une lumière rouge.

3.3.1 Description des sources lumineuses

Une source lumineuse peut être *directionnelle* ou *locale*. Dans le premier cas, elle est considérée comme placée à l'infini, les rayons émis sont alors supposés parallèles, et l'ensemble de la scène est éclairée de façon uniforme. Dans ce cas, seule sa couleur et sa direction d'éclairage sont spécifiées. Le soleil est un bon exemple de source directionnelle. Si la figure 9 nous montre bien l'effet d'une lumière directionnelle, on remarquera également l'absence d'ombrage porté par le cube situé entre la source lumineuse et la plante. Le cas de l'ombre portée sera abordé dans un paragraphe ultérieur (Cf.3.5).

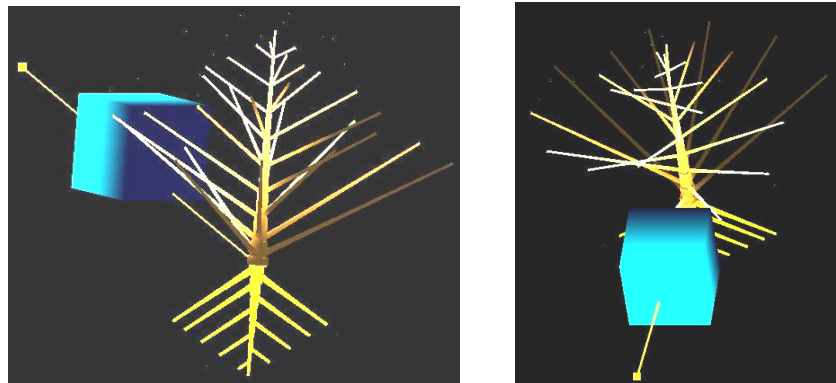


Figure 9 Objets éclairés par une source lumineuse directionnelle.

Les sources locales sont utilisées pour rendre compte d'objets éclairants tels les lampes ou les lampadaires (Fig.10). Elles sont caractérisées par leur position dans l'espace et par la direction de leur éclairage. En plus de leur couleur, on spécifie un cône d'éclairage dans lequel sont émis les rayons lumineux (plus l'angle d'ouverture du cône est faible, moins l'espace éclairé est grand) et un facteur d'atténuation de la lumière (plus le rayon s'éloigne de son point d'émission, moins il est intense).

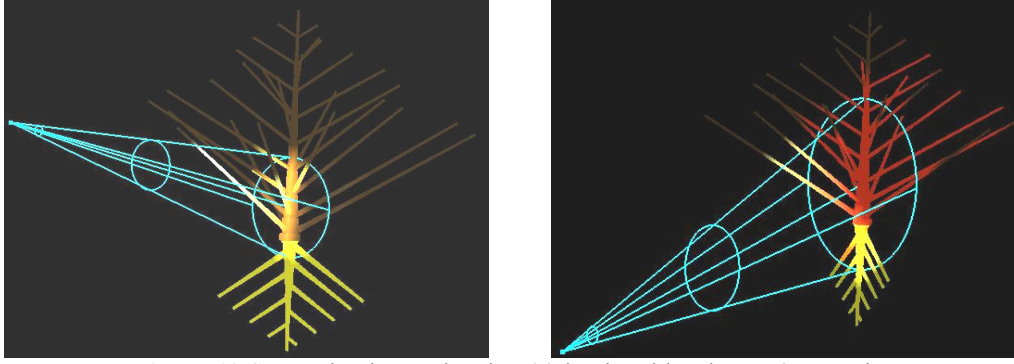


Figure 10 Source lumineuse locale : (a) lumière blanche et cône étroit ,
(b) lumière rouge et cône plus large.

3.3.2 Modèles de couleur

Afin de caractériser les couleurs, des modèles de couleur ont été définis. Ces modèles spécifient un système de coordonnées 3D dans lequel un sous-espace décrit les différentes teintes (Fig.11) :

- Le modèle rouge, vert, bleu (RGB) utilise un système de coordonnées cartésien (Fig.11.a). Les couleurs primitives du modèle RGB sont dites additives : chaque contribution individuelle d'une couleur primitive s'ajoute aux autres pour donner la teinte finale.
- Le modèle *HSV* (*hue, saturation, value* ou teinte, saturation, valeur) est défini par une pyramide à 6 faces, elle-même décrite dans un système de coordonnées cylindriques (Fig.11.b). On trouve les couleurs les plus brillantes au sommet de la pyramide pour lesquelles le paramètre $V=1.0$. La teinte (H) est donnée par l'angle autour de l'axe vertical. La saturation varie de 0.0 le long de l'axe vertical à 1.0 sur les facettes de la pyramide.
- Le modèle *HLS* (*hue, lightness, saturation* ou teinte, brillance, saturation) est représenté sous la forme d'une double pyramide à 6 faces dans un sous-espace cylindrique (Fig.11.c). La teinte (H) est définie par l'angle autour de l'axe vertical de la double pyramide. La brillance (L) est de 0.0 pour la couleur noire à la base de la double pyramide et 1.0 pour la couleur blanche à son sommet. Les teintes grises ont toutes une saturation (S) à 0.5 , mais la saturation maximale des teintes est obtenue pour les valeurs $S=1.0$ et $L=0.5$.

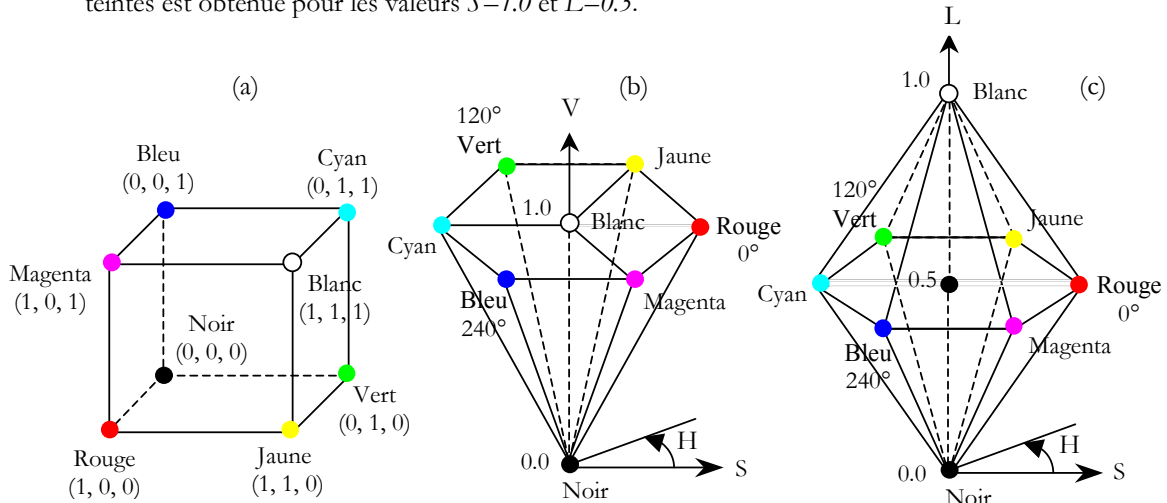


Figure 11 Modèles de couleur : (a) RGB, (b) HSV, (c) HLS.

D'autres modèles de couleur, que nous ne présenterons pas ici, ont été construits : *CMY* (*cyan, magenta, yellow*), *YIQ* (luminance dans Y et couleur dans I et Q).

3.3.3 Modèles d'illumination et propriétés des matériaux

Le modèle d'illumination le plus simple consiste à placer les objets dans un espace sans source lumineuse où leur couleur sera calculée en fonction d'une intensité qui leur est propre. Ce modèle, appelé *global ambient light*, affiche des objets monochromatiques dans un environnement sans réflexion (Fig.12.b). Si, maintenant, on imagine une source de lumière diffuse et non directionnelle, la couleur des objets résultera des réflexions multiples entre les différentes faces des objets présents. Ce phénomène est appelé *ambient light* (Fig.12.c). La quantité de lumière réfléchie par la surface d'un objet est déterminée par son coefficient *ambient-reflection*. Ce coefficient caractérise le matériau constituant l'objet et définit donc une de ses propriétés.

Nous venons de voir que les objets illuminés par la lumière ambiante sont plus ou moins brillants proportionnellement à l'intensité ambiante, mais sont uniformément éclairés le long de leurs surfaces. Si

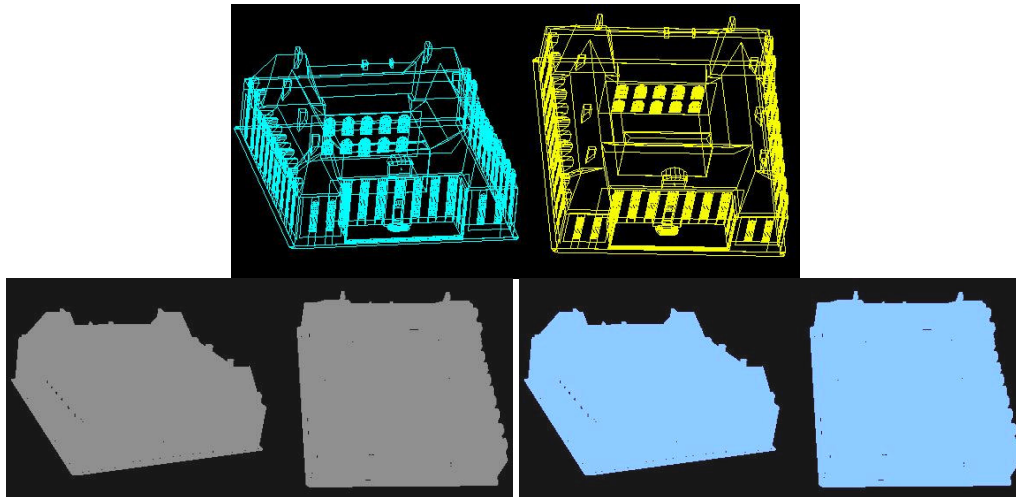


Figure 12 Modèle de lumière ambiante : (a) modèle en filaire, (b) global ambient light, (c) ambient light.

on considère une source de lumière ponctuelle dont les rayons sont émis uniformément dans toutes les directions, la brillance de l'objet variera d'un point à un autre suivant la distance le séparant de la source et l'angle formé par la direction de la lumière et la normale à la surface au point considéré. Ce modèle est appelé *réflexion diffuse* (Fig.13). Des surfaces apparaissent mates ou ternes quand elles ont le même éclat quel que soit le point de vue. Ceci s'explique par le fait qu'elles réfléchissent la même quantité de lumière dans toute les directions. Le *coefficient de réflexion diffuse* est une autre propriété du matériau qui permet de rendre compte de sa nature.

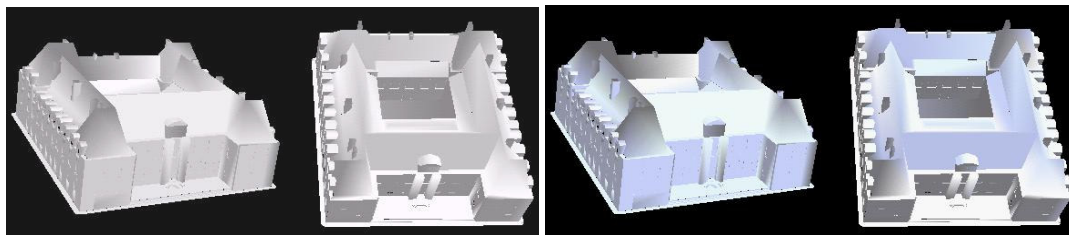


Figure 13 Modèle de réflexion diffuse.

Si on éclaire un objet par un rayon de lumière blanche intense, l'aspect brillant qui en résulte au point d'incidence est dû à la *réflexion spéculaire* de l'objet (Fig.14), alors que la lumière réfléchie par les autres parties de l'objet provient de la réflexion diffuse. Au point d'incidence du rayon lumineux, l'objet apparaîtra de la couleur du rayon et non de celle d'origine. Si on modifie le point d'observation, on remarque que la partie rendue brillante de l'objet change également. Ceci s'explique par le fait que les surfaces brillantes réfléchissent la lumière non uniformément dans toutes les directions. L'*exposant de*

réflexion spéculaire, autre caractéristique du matériau, permet de moduler l'aspect de la réflexion spéculaire : une valeur faible (de l'ordre de 1) décrit une réflexion large et diffuse, alors qu'une valeur plus élevée (de l'ordre de plusieurs centaines) spécifie une réflexion très focalisée.

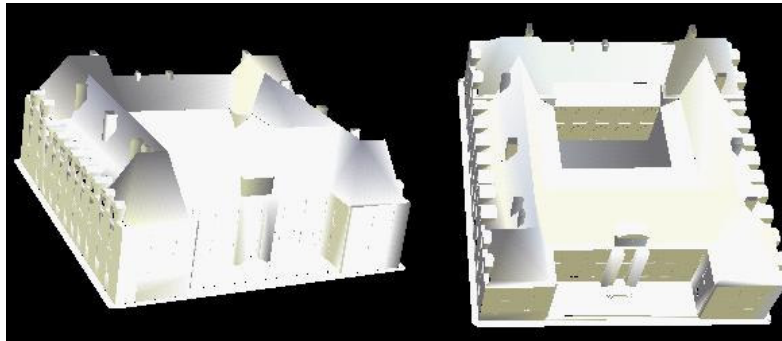


Figure 14 Modèle de réflexion spéculaire.

3.3.4 Interpolation des couleurs

L'interpolation des couleurs (*interpolated shading*) permet de calculer pour chaque sommet d'un polygone une couleur qui servira ensuite à interpoler la couleur de chaque pixel composant le polygone. L'interpolation permet de tenir compte avec plus de précision des variations dues à l'ombrage sur une face. L'amélioration apportée est d'autant plus nette que l'objet est de forme courbe plutôt que réellement polyédrique (toutes les faces le composant sont planes).

Plusieurs méthodes ont été développées pour effectuer ce type d'interpolation :

- Le *Gouraud shading* (Gouraud, 1971) interpole les intensités lumineuses, éliminant ainsi les discontinuités (Fig.15).
- Le *Phong shading* (Bui-Tong, 1975) interpole les normales aux surfaces plutôt que les intensités. Ces normales sont elles-mêmes interpolées le long des côtés des surfaces à partir des normales aux points, comme dans le Gouraud shading.

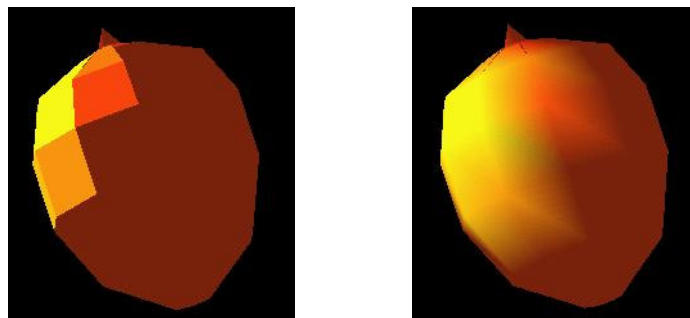


Figure 15 Interpolation des couleurs: (a) sans Gouraud shading ,
(b) avec Gouraud shading .

3.4 APPLICATION DE TEXTURES

Il est difficile et coûteux en temps calcul et en mémoire de décrire dans le détail les particularités de chaque surface. Une solution est de plaquer une image digitalisée ou de synthèse sur la surface considérée. Cette technique, initiée par Catmull (1974) puis améliorée par Blinn et Newell (1976), est connue sous le nom de *plaquage de texture* (*texture mapping*) (Fig.16).

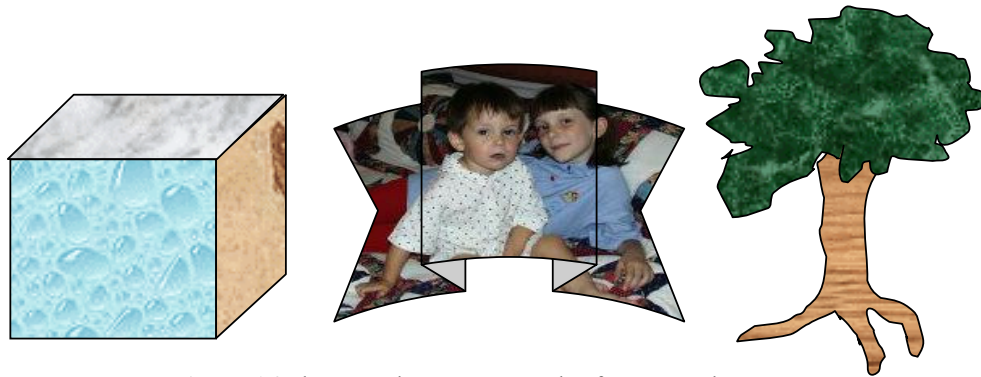


Figure 16 Plaquage de textures sur des formes quelconques.

3.5 OMBRES PORTEES

Si l'on tient compte de l'ombrage que portent les objets d'une scène sur les autres objets de la même scène, le réalisme final donné à l'image générée en sera encore accru.

Les algorithmes d'élimination des parties cachées (Cf.3.2) déterminent quelles sont les faces visibles des objets de la scène depuis le point de vue de l'observateur ; les algorithmes de calcul des ombres portées déterminent quelles faces peuvent être vues depuis la source lumineuse. On comprend alors que les deux types d'algorithmes sont fondamentalement semblables. Les faces visibles depuis la source lumineuse sont éclairées alors que celles non visibles sont dans l'ombre. Si on a défini plusieurs sources lumineuses, on peut faire le même test pour chaque face à partir de chaque source lumineuse.

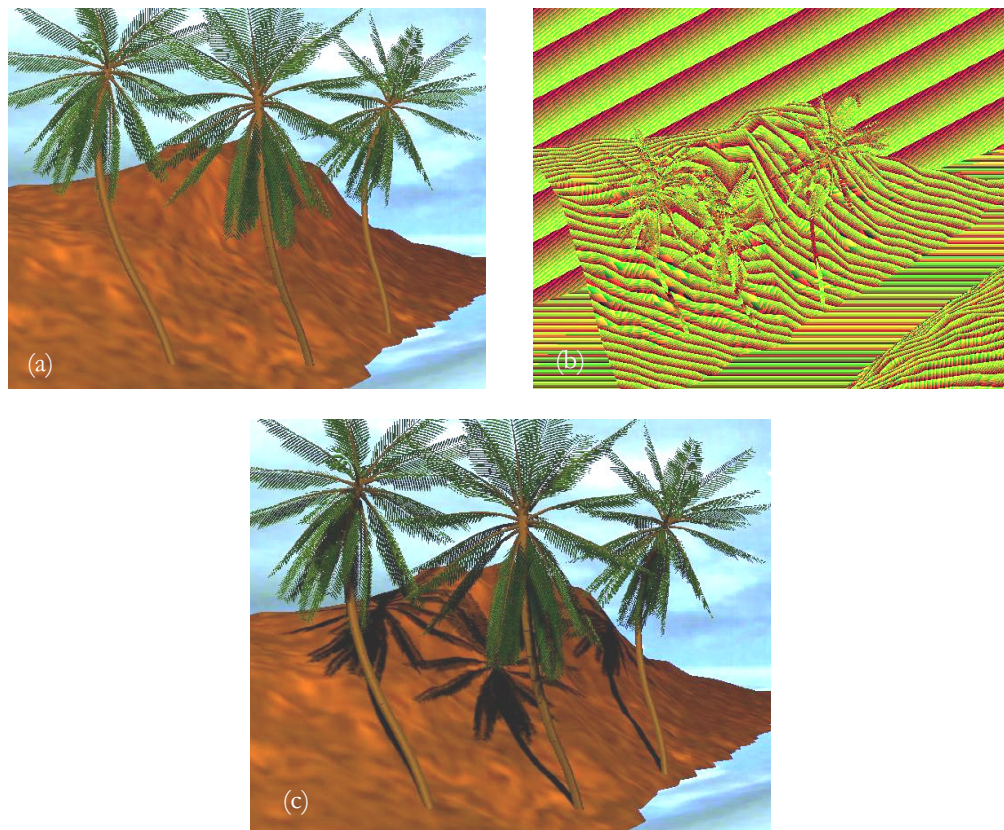


Figure 17 Technique du double z-buffer : (a) vue du point d'observation, (b) vue de la source lumineuse en fausse couleur, (c) image finale avec calcul des ombres portées.

Un des algorithmes développés dans ce but est connu sous le nom de *double z-buffer*. Comme son nom l'indique, il est basé sur la technique du z-buffer (Cf.3.2) mais au lieu d'effectuer un unique traitement, il en applique deux : un à partir de la source lumineuse (Fig.17.a) et un second depuis le point de vue de l'observateur (Fig.II.17.b). Le principe en est le suivant :

1. Calcul et stockage du z-buffer pour l'image à partir de la source lumineuse.
2. Calcul et stockage du z-buffer et de l'image à partir du point de vue de l'observateur après les modifications qui suivent :
 - Si un pixel est détecté comme visible, ses coordonnées (x, y, z) dans le repère de l'observateur sont transformées en coordonnées (x', y', z') dans le repère de la source lumineuse.
 - Les coordonnées transformées x' et y' sont utilisées pour sélectionner la valeur z_L dans le z-buffer de la source lumineuse qui doit être comparée à la valeur transformée z' .
 - Si z_L est plus près de la source que z' , la présence d'un obstacle entre la source et le point a été détectée et le point est dans l'ombre. Dans le cas contraire, le point est visible depuis la source lumineuse et est donc éclairé.

La figure 17.c nous montre le résultat d'un calcul d'ombre portée réalisé par la méthode du double z-buffer.

3.6 TRANSPARENCE ET REFLEXION

Si la plupart des surfaces ont des réflexions diffuse et spéculaire, certains matériaux qui transmettent la lumière peuvent être transparent comme le verre ou simplement translucide.

Un rayon lumineux traversant une surface transparente est normalement déviée de sa trajectoire initiale. Cette déviation est appelée *réfraction*. La relation entre l'angle d'incidence θ_i du rayon lumineux et l'angle de réfraction θ_t est donnée par la formule de Snell :

$$n_i \sin \theta_i = n_t \sin \theta_t$$

où n_i et n_t sont les indices de réfraction des matériaux à travers lesquels est passée le rayon lumineux.

On peut cependant modéliser la transparence sans tenir compte du phénomène de réfraction mais on perd en réalisme. La figure 18 nous expose les deux cas de figure. Si on ne dévie pas le rayon lumineux au passage de l'objet transparent, le cube rouge est visible, alors, qu'en cas de réfraction du rayon, c'est le cube bleu qui est visible.

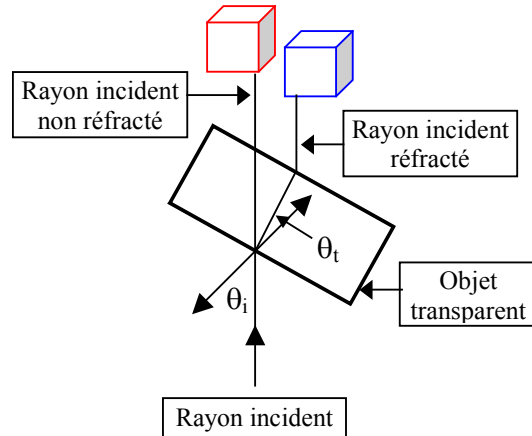


Figure 18 Réfraction d'un rayon lumineux.

De même que la réfraction ajoute du réalisme aux images générées, un modèle de la réflexion permet de simuler les formes réfléchies par un miroir ou par d'autres objets de la scène moins réfléchissants. Un rayon réfléchi a un angle de réflexion égal à l'angle d'incidence par rapport à la normale au point d'incidence.

La figure 19 résume les lois de la réfraction et de la réflexion.

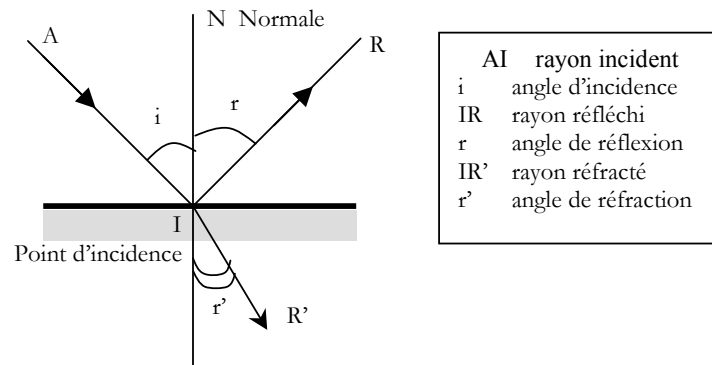


Figure 19 Réflexion et réfraction.

3.7 DES MODELES D'ILLUMINATION GLOBALE

Un modèle d'illumination calcule la couleur en un point en considérant la lumière directement émise par des sources lumineuses (appelée *illumination locale*) et la lumière provenant d'autres surfaces par réflexion ou transmission (appelée *illumination globale*). Dans un précédent paragraphe (Cf.3.3.3), nous avons vu que l'utilisation d'un modèle d'illumination ambiant permettait de simuler l'illumination globale mais en affectant la même luminosité à tout point de tout objet de la scène : la luminosité ne dépend ni de la position des objets, ni de la position de l'observateur, ni d'éventuels obstacles entre la source lumineuse et le point considéré. Par ailleurs, nous avons présenté quelques modèles susceptibles de rendre compte de certains effets d'illumination : ombres portées (Cf.3.5), transparence et réflexion (Cf.3.6).

Afin d'améliorer le calcul de l'illumination globale dans une scène (dans un environnement naturel, l'illumination globale est plus importante que l'illumination locale), deux différentes classes d'algorithmes ont été développées :

1. les algorithmes de lancer de rayons (*ray-tracing*) (Fig.20) : ils traitent en même temps l'élimination des parties cachées et le calcul de la couleur en tenant compte des ombres portées, des réflexions et des réfractions (Whitted, 1980). Bien que la méthode du lancer de rayon se soit montrée très efficace dans la modélisation de la réflexion spéculaire, elle nécessite la définition d'une lumière ambiante globale.
2. les algorithmes de radiosité (Fig.21) : contrairement au lancer de rayon, ces méthodes séparent complètement le calcul des couleurs et la détermination des faces cachées (Goral *et al.*, 1984 ; Sillion, 1995). Les approches basées sur des modèles d'échanges thermiques pour l'émission et la réflexion radiatives éliminent cette contrainte en traitant de façon plus adéquate les réflexions inter objets. Ces algorithmes supposent une conservation de l'énergie lumineuse dans un environnement clos : toute énergie émise ou réfléchie par chaque surface est prise en compte dans la réflexion sur ou l'absorption par les autres surfaces. Contrairement aux algorithmes de rendu classiques, les méthodes de radiosité déterminent, dans un premier temps, toutes les interactions lumineuses dans un environnement indépendant de la position de l'observateur, puis, dans un second temps, génèrent l'image en ne traitant que les faces cachées et l'interpolation des couleurs.



Figure 20 Rendu par lancer de rayons.

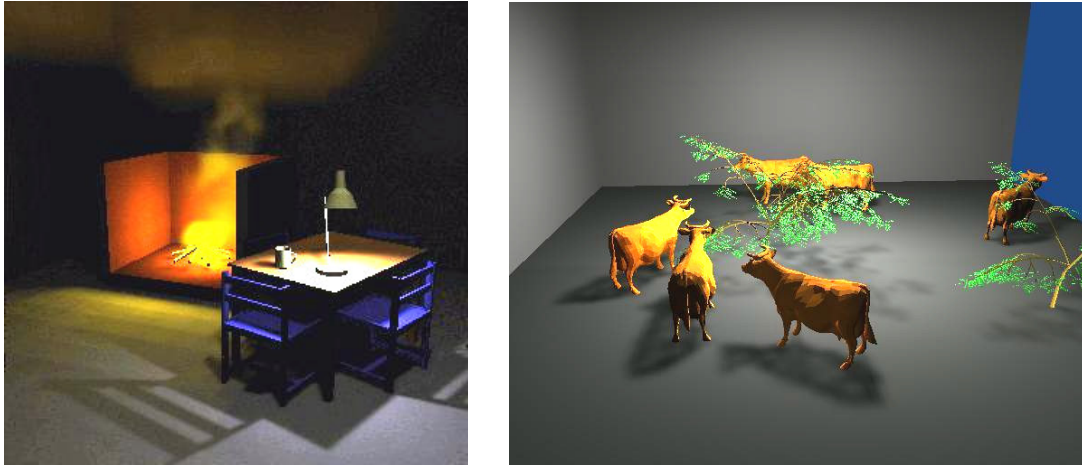


Figure 21 Rendu par radiosit  (F.Sillion, i MAGIS).

3.8 AFFICHAGE D'UNE SCENE 3D

Visualiser une sc ne 3D sur un p riph rique 2D (g n ralement un  cran) n cessite de d finir les  l ments suivants :

- un volume de visualisation dans l'espace r el o  ont  t  d crits les diff rents objets constituant la sc ne,
- une projection sur un plan de projection,
- une zone d'affichage sur le p riph rique consid r .

Le volume de visualisation est d fini par un plan de coupage avant (*front clipping plane*) et un plan de coupage arri re (*back clipping plane*), chacun positionn  par rapport   l'observateur (Fig.22). La projection peut  tre de diff rents types. En particulier, elle peut  tre parall le (Fig.22.a) ou perspective (Fig.22.b). Dans ce dernier cas, la forme du volume de visualisation est une pyramide tronqu e.

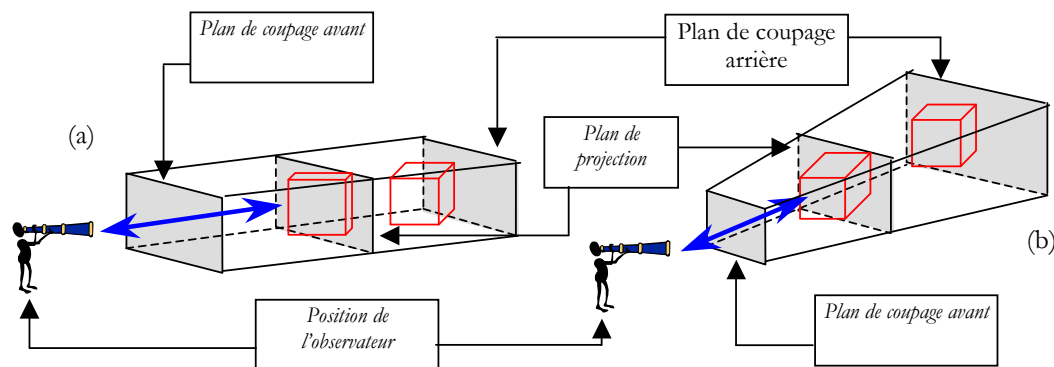


Figure 22 D finition du volume de visualisation et du plan de projection :
(a) projection parall le, (b) projection perspective.

Pratiquement, les objets de la sc ne sont coup s par le volume de visualisation puis projet s sur le plan de projection. Le r sultat de la projection (Fig.23) est alors affich  sur le p riph rique d'affichage dans la zone d finie.

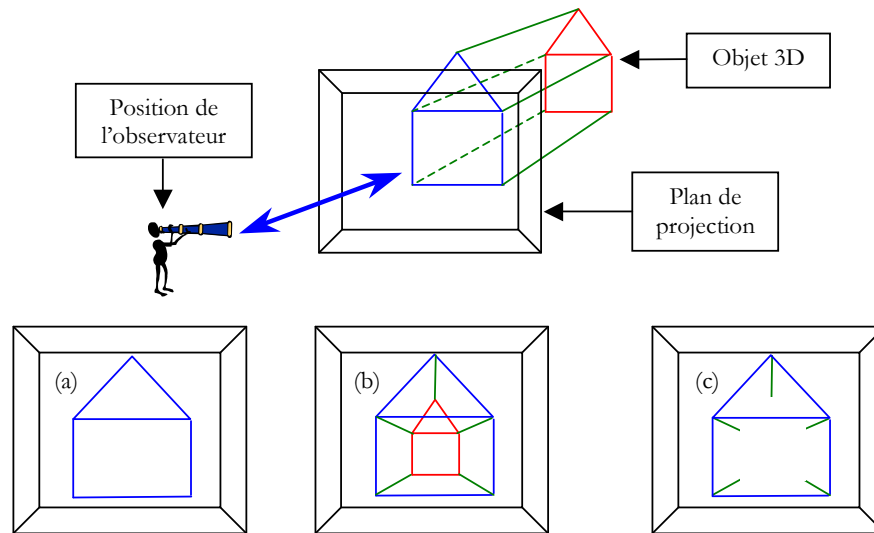


Figure 23 Projection d'un objet 3D : (a) parallèle, (b) perspective, (c) perspective avec le plan de coupage arrière situé entre l'observateur et la face arrière de la maison (en rouge).

4 METHODES INFORMATIQUES POUR LA SIMULATION ET LA REPRESENTATION DES PLANTES VIRTUELLES

4.1 DES MODELES INFORMATIQUES POUR LA REPRESENTATION DES PLANTES

L'apport de l'informatique dans le domaine de la modélisation des plantes concerne notamment la simulation de la dynamique de croissance. En effet, l'informatique introduit la notion de génération algorithmique des formes, ce qui permet de ne plus se limiter à une description statique de propriétés géométriques.

Nous allons présenter ici les principaux modèles informatiques utilisés dans le domaine de la modélisation et de la représentation des plantes. Notons que certains auteurs ont déjà pratiqué ce genre d'exercice (Fournier, 1987 ; Françon, 1990). Il nous paraît cependant intéressant de répéter l'expérience en mettant l'accent sur le contexte plus ou moins botanique dans lequel ont été développées les différentes méthodes, et ce afin de bien mettre en évidence les différents types de résultats que l'on peut obtenir suivant le modèle utilisé.

4.1.1 Des modèles de plantes figuratives

J.B. Fisher et H. Honda (Honda, 1971 ; Fisher et Honda, 1979a et 1979b) définissent une plante par un nombre restreint de paramètres géométriques tels les angles de branchement et les longueurs des tiges. Le principe général en est simple : chaque branche se sépare en deux autres branches, l'opération de ramification s'arrêtant à un niveau donné. A chacun des branchements interviennent trois angles, les deux angles de branchement, et un angle dit de divergence pour tenir compte de la phyllotaxie, et deux rapports de diminution de la longueur des branches par rapport à la branche porteuse. Ce modèle purement géométrique permet d'engendrer des plantes différentes dans leur géométrie mais pas dans leur topologie, d'ailleurs très succincte (Fig.24). L'objectif ici est bien une approche botanique de la plante ; on ne cherche pas seulement à créer des images.

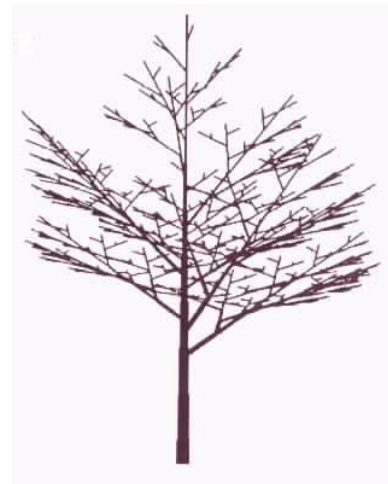


Figure 24 Modèle de Honda.

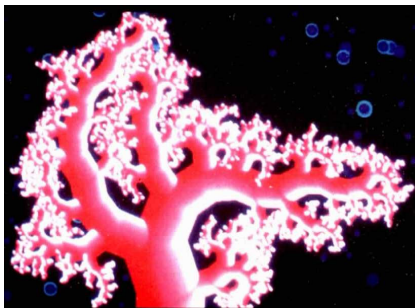


Figure 25 Plante spiralee.

Kawaguchi (1982), inspiré en partie par les travaux de D'Arcy Thompson (1966) sur la géométrie des formes dans la nature, produit des images figuratives de formes naturelles: coquillages, cornes, plantes spiralees (Fig.25). Il ne cherche cependant pas à reproduire la réalité mais à produire des images aux formes naturelles variées et complexes.

Gardner (Gardner, 1984) décrit une méthode de génération rapide de feuillage orientée vers la synthèse d'images animées. La méthode consiste à approximer le feuillage par une forme géométrique, une conique en général, puis à

générer une texture qui sera ensuite plaquée sur la forme. Ici, la plante est représentée dans sa forme globale, seule la texture tend à simuler le détail. On obtient alors par cette méthode des plantes figuratives.

Un objet fractal est défini par Mandelbrot (Mandelbrot, 1977 et 1982) comme un objet qui montre d'autant plus de détails que l'on observe sa forme avec un facteur de grossissement plus fort. Si, à n'importe quel niveau de détail, l'objet est identique dans son apparence, alors, on parle d'objets *auto-similaires* (Fig.26).

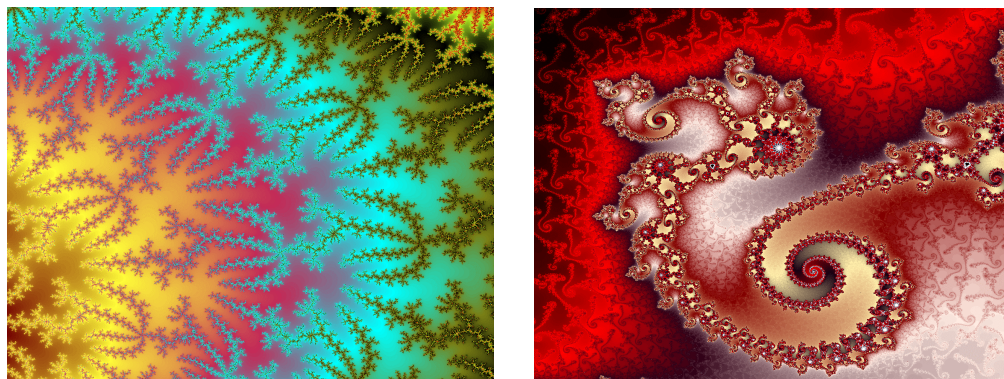


Figure 26 Objets fractals : on remarquera l'auto-similarité.

C'est en se basant sur les théories fractales de Mandelbrot, et sur le postulat selon lequel la notion géométrique d'auto-similarité fractale est une loi générale pour les formes dans la nature, que Oppenheimer a développé un modèle fractal générant des objets ramifiés : plantes (Fig.27), feuilles, réseaux artériels, ou encore, deltas fluviaux et flocons de neige (Oppenheimer, 1986). La topologie des plantes est assurée par un modèle récursif, et la géométrie par plusieurs paramètres : angle de branchement, rapport de longueur entre les branches, nombre de ramifications. Les formes végétales générées sont auto-similaires.



Figure 27 Plante fractale.

Mais, pour représenter la variabilité naturelle, à chaque paramètre géométrique sont associées une valeur moyenne et une valeur de déviation. Les plantes ainsi obtenues ne sont donc que statistiquement auto-similaires. Si les images obtenues par ce modèle sont de très belle qualité, on peut toutefois s'interroger sur la validité du postulat énoncé, tout du moins en ce qui concerne les plantes. En effet, peu de plantes dans la nature semblent auto-similaires, sauf quelques plantes primitives comme les fougères. L'intérêt de cette méthode se limite sans nul doute au domaine de la représentation de la plante en infographie.

Aono et Kunii se sont intéressés aux L-systèmes (Cf.4.1.2) pour représenter le mode de branchement des plantes (Aono et Kunii, 1984). Mais ils ont jugé que l'ajout de règles de production permettant des manipulations géométriques et des applications graphiques était trop compliqué. Aussi, ils ont défini un modèle géométrique en 3D appelé GMT1 (Fig.28), qui retombe dans les carences du modèle de Fisher et Honda, à savoir une pauvreté et une rigidité de la topologie. On s'éloigne à nouveau de la réalité botanique. Ils l'ont cependant complété par d'autres modules qui améliorent la souplesse de la simulation :

- GMT2: simulation du vent, de la lumière ou de la gravité par placement de points attractifs ou

répulsifs,

- GMT3: la ramification passe de binaire à ternaire (Fig.29),
- GMT4: les angles de branchement ne sont plus constants mais varient en fonction du niveau de croissance de la plante.

L'ensemble de ces modules forme le *A-système*, un système interactif de manipulations de plantes de synthèse.



Figure 28 Modèle GMT1 de Aono et Kunii.

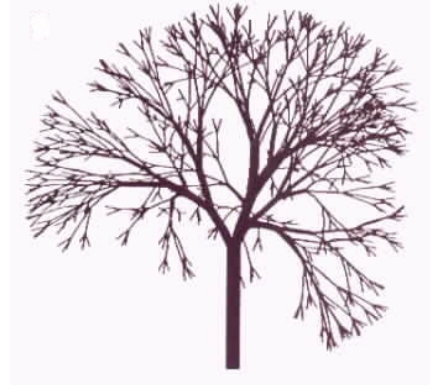


Figure 29 Modèle GMT3 de Aono et Kunii.

La notion de système de particules proposée par W.T. Reeves définit le volume de l'objet (Reeves, 1983). Au bout d'un certain laps de temps, des particules sont générées à l'intérieur du système, elles se déplacent, changent de forme et finissent par mourir. La représentation de systèmes de particules diffère



Figure 30 Génération d'une forêt par la méthode du système de particules.

en trois points des représentations habituelles. D'abord, un objet n'est pas représenté par un ensemble de surfaces de base, comme des polygones, qui définissent ses frontières, mais par des nuages de particules qui définissent son volume. Deuxièmement, un système de particules n'est pas une entité statique. Ses particules changent de forme et se meuvent dans le temps. Finalement, les caractéristiques d'un objet représenté par un tel système peuvent être stochastiques. Ainsi, son utilisation a permis à Reeves et Ricki Blau de générer une forêt (Fig.30), après quelques aménagements cependant (Reeves et Blau, 1985).

En effet, chaque plante est

représentée comme un ensemble de segments linéaires, ou même de petits cercles, qui constituent ses branches et ses feuilles. De nombreuses relations complexes existant entre les particules représentant les branches et les feuilles, celles-ci doivent former un objet 3D cohérent. Le système doit donc générer des formes plus structurées et les particules doivent être interdépendantes. Les images obtenues sont de belle facture mais ne peuvent intéresser le botaniste.

Une nouvelle méthode de génération de formes fractales a été mise au point par M.F. Barnsley, et sera utilisée par d'autres chercheurs. On appelle *Système de Fonctions Itératives* ou *SFI* (Barnsley, 1988), tout ensemble de transformations affines (rotation, translation ou modification de l'échelle) du plan sur lui-même auxquelles on a affecté une probabilité. Ces fonctions doivent être *contractives*: le facteur de la fonction de modification de



Figure 31 Fougère fractale générée par un SFI.

l'échelle est inférieure à 1.0. Chaque SFI possède un unique point d'attraction qui est un objet fractal (Fig.31). Le design d'objets fractals est relativement facile à réaliser et intuitif, grâce à la découverte d'une importante propriété mathématique faisant le lien entre les fractales et les SFI. Cette méthode permet également de résoudre le problème inverse : donner la géométrie d'un objet et déterminer un SFI qui générera approximativement cette géométrie (Persi et Shahshahani, 1984). Cette méthode se caractérise par les mêmes défauts déjà évoqués pour le modèle défini par Oppenheimer, défauts inhérents au concept géométrique d'auto-similarité fractale. On reste dans le domaine de la plante figurative.

En s'intéressant plus particulièrement à un arbre, l'érable, J. Bloomenthal a décrit un modèle basé principalement sur la géométrie. La plante est décrite par des coordonnées 3D et par les connexions reliant ces points (Bloomenthal, 1985). La forme du tronc et des branches est assurée par des cylindres dont les axes sont des courbes dans l'espace qui interpolent ces points. Quant aux feuilles, elles se composent de surfaces simples sur lesquelles sont appliquées des textures. Si le résultat est très réussi au niveau visuel (Fig.32), il n'est pas question de modifier aisément la topologie de la plante, et encore moins de simuler un quelconque processus de développement. L'image est de nouveau à l'honneur.



Figure 32 Erable avec plaquage de textures.

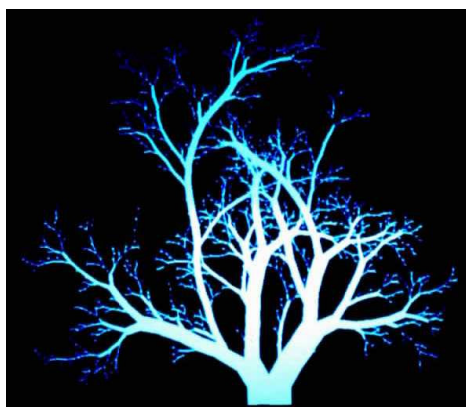


Figure 33 Arbre combinatoire.

Une autre approche est utilisée par Jean Françon, Xavier Viennot et leurs élèves pour définir la topologie de la plante. En se basant sur l'analyse de Horton et Strahler (Horton, 1945 ; Strahler, 1952) pratiquée en hydrogéologie, et les travaux sur les arbres en combinatoire et leurs applications dans différents domaines (Françon, 1984 ; Eyrolles *et al.*, 1986 ; Eyrolles, 1986 ; Aries et Viennot, 1990 ; Viennot, 1990), on définit la forme de plantes binaires, puis on réalise leur géométrie en introduisant les angles de branchement, la longueur et le diamètre des branches. Les auteurs spécifient bien que leur modèle se restreint à la représentation de plantes figuratives (Fig.33).

En marge de ces modèles de croissance de plantes, on peut également évoquer des travaux informatiques qui cherchent à générer des surfaces de formes naturelles et très variées, formes utilisées notamment pour représenter des feuilles, des fleurs ou des fruits (Lienhardt, 1987 ; Lienhardt et Françon, 1987).

Tous les modèles que nous venons de présenter ont le mérite de proposer des images le plus souvent remarquables. Toutefois, malgré un certain effort pour intégrer une réalité botanique dans les plantes simulées, le botaniste ne pourra y trouver son compte.

« Il ne s'agit pas d'imiter la nature mais de travailler comme elle. »

Pablo Picasso (Quéau, 1986)

4.1.2 Les L_systèmes⁷

Dans le but de décrire la croissance d'organismes vivants, Lindenmayer (Lindenmayer, 1968a et 1968b) a introduit la notion de systèmes de réécritures parallèles. Il s'agit de grammaires définies par un alphabet et par des règles de réécriture ou de production. L'évolution d'un tel système se fait en générant des mots créés à partir d'une lettre de l'alphabet ou d'un mot précédemment généré. La simultanéité de l'application des règles de production en vigueur dans les L-systèmes a été conçue pour représenter la simultanéité de développement de parties d'organismes biologiques. Elle semblerait donc pouvoir répondre de façon satisfaisante à la formalisation et à la simulation de la croissance des plantes. Ainsi, ces L-systèmes ont été utilisés afin de définir les structures de branchement des plantes. Initialement, les L-systèmes ne supportaient pas la définition de règles graphiques. Les premières études de l'application des L-systèmes aux images ont été menées par Fritjters et Lindenmayer (1974) et Hogeweg et Hesper (1974). Ces derniers ont ainsi obtenu des images, mais sans grand réalisme. Il faut attendre les travaux de A.R. Smith pour voir les L-systèmes effectivement exploités pour la génération de plantes réalistes. Ainsi, la notion de *graftal* fait son apparition. L'auteur définit un objet graftal comme le résultat topologique d'une grammaire formelle, en l'occurrence un L-système, géométriquement interprété (Smith, 1984). Cette méthode oblige l'utilisateur à changer de règles de production pour obtenir une plante différente. De plus, la dynamique de croissance des végétaux n'est pas vraiment prise en compte par une telle simulation. Le réalisme des plantes obtenues se restreint à l'aspect visuel.

Prusinkiewicz continue de développer à la suite des travaux de Smith l'interprétation graphique des L-systèmes. Ainsi, il a développé un modèle qui permet d'engendrer une séquence de symboles, qu'il utilise ensuite pour contrôler les mouvements d'une *tortue* qui trace l'image. En travaillant en particulier sur les plantes herbacées, Prusinkiewicz, Lindenmayer et Hanan améliorent les bases botaniques du modèle, et tente de simuler le processus de développement de la plante (Prusinkiewicz *et al.*, 1988). Cette nouvelle approche représente sans nul doute une évolution importante du concept de simulation des plantes en utilisant des L-systèmes.

Si les travaux de Prusinkiewicz ont toujours fait référence en infographie dans le domaine de la modélisation de la croissance et de la représentation des plantes, les applications à l'agronomie ne sont apparues que plus tard, notamment grâce aux travaux de plusieurs de ses collègues (Room et Hanan, 1994 ; Hanan, 1995 ; Room *et al.*, 1996). Toutefois, grâce à quelques illustrations provenant de ses recherches, nous verrons que son souhait est bien de développer des modèles utilisables par les biologistes.

« I was no longer happy with pretty pictures. I wanted plants based on nature. »

Przemyslaw Prusinkiewicz (Robertson, 1989)

4.1.2.1 La théorie

Afin de simuler des phénomènes endogènes, Aristide Lindenmayer a proposé un modèle de développement basé sur des *règles de réécriture*, encore appelées *règles de production* (Lindenmayer, 1968a et 1968b). A l'origine, ce modèle, connu sous le nom de L-systèmes, permettait une description formelle du développement d'organismes multicellulaires simples, mais il a ensuite été appliqué à des plantes plus grandes et plus complexes.

Les règles de production opèrent sur des modules entiers d'une plante (un bourgeon, un entre-nœud ou une feuille par exemple). Chaque règle est

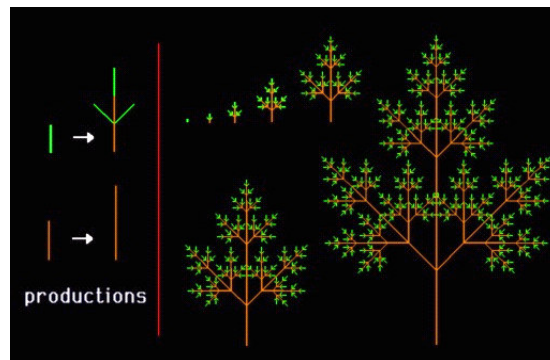


Figure 34 Génération d'une structure ramifiée à l'aide d'un L-système (Prusinkiewicz, 1993).

⁷ D'après Prusinkiewicz *et al.*, 1994

composée de deux entités : un prédécesseur et un successeur. Pendant une étape de dérivation, le prédécesseur, identifié par son label, est remplacé par son successeur. Lindenmayer a développé une notation sous forme de chaînes de caractères qui permet de spécifier simplement les règles de production et qui autorise la simulation. Ainsi, même des L-systèmes très simples peuvent générer des structures ramifiées semblables à celles des plantes. La figure 34 nous montre le résultat obtenu par simulation d'un tel L-système. La première règle indique que l'apex (en vert) génère une sous-structure ramifiée composée de 3 apex et d'un entre-nœud. La seconde règle augmente la longueur de l'entre-nœud. Le processus de développement débute avec un apex et finit par produire une structure de feuille composée comme on peut en voir chez certaines fougères.

4.1.2.2 Quelques applications des L-systèmes par P. Prusinkiewicz

En nous appuyant sur des images issues de ses recherches, nous allons présenter succinctement quelques-unes des réalisations de Prusinkiewicz. Si des L-systèmes très simples permettent de générer des structures ramifiées, d'autres aussi simples sont capables de produire des inflorescences de plantes herbacées (Prusinkiewicz, 1988) (Fig.35, 36).



Figure 35 Inflorescences de lilas.

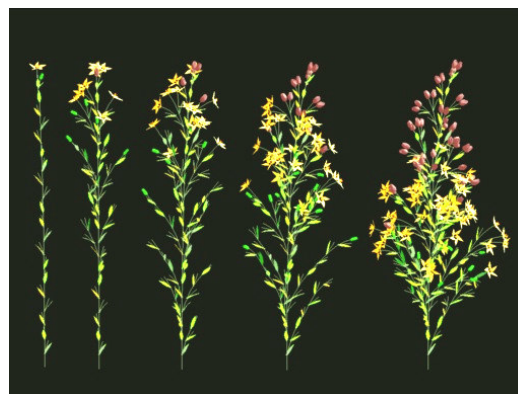


Figure 36 Simulation d'une séquence de floraison.

Pour rendre compte de la disposition des organes dans une plante, Prusinkiewicz introduisit un modèle spécifique de phyllotaxie spiralée (Prusinkiewicz et Lindenmayer, 1990) (Fig.37 et 38).

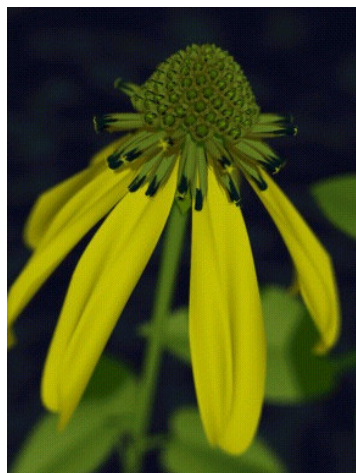


Figure 37 *Rudbeckia laciniata*.



Figure 38 Cactus en fleur.

Les L-systèmes *sensibles à l'environnement* ont permis à Prusinkiewicz de simuler l'effet de la taille sur la forme générale des plantes. Dans ce modèle, les plantes sont supposées croître dans un espace limité. Si un bourgeon en se développant atteint les limites de cette espace, il cesse sa croissance et des bourgeons jusqu'alors dormants prennent le relais : on peut parler ici de simulation de réitérations (Fig.39). Cette

méthode a été en particulier utilisée pour créer des jardins où on simule la taille des plantes en les confinant dans des volumes géométriques aux formes variées (Fig.40).

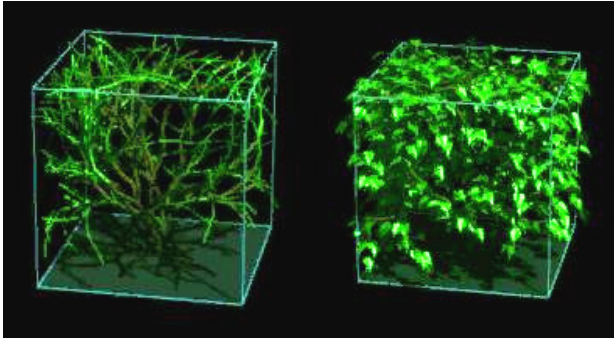


Figure 39 Développement d'un système ramifié dans un espace limité : sans feuille et avec feuilles.

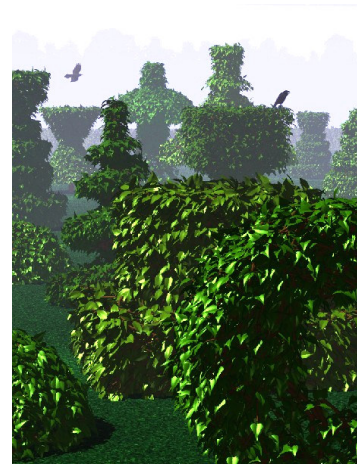


Figure 40 Plantes taillées.

Si la taille mécanique peut être considéré comme un phénomène exogène influençant la croissance de la plante, il en existe d'autres plus complexes à modéliser : effet de la lumière, compétition pour l'espace entre plantes voisines ou au sein d'une même plante. En particulier, si ces conditions environnementales influent sur la croissance de la plante, celle-ci modifie également l'environnement en occupant un espace géométrique toujours différent tout au long de son développement. Les *L-systèmes ouverts* permettent de simuler les interactions plantes/environnement (Mech et Prusinkiewicz, 1996). Ici, les plantes se développent dans un espace caractérisé par des scalaires ou un champ de vecteurs. En chaque point de l'espace, les valeurs spécifiées pourront éventuellement modifier la stratégie de croissance de la plante.

La figure 41 reprend l'expérience de Honda (1971) sur la croissance de branches de *Terminalia catappa* en spécifiant des gradients de densité. Il s'agit ici de détecter et d'éviter les collisions entre branches. La figure 42 montre le développement de deux systèmes racinaires influencés par la densité hydrique du sol. La quantité et la localisation de l'eau disponible dans le sol sont prédéterminées. Au cours de la croissance, les racines et les racinelles cherchent les zones irriguées, absorbent l'eau et cessent de croître si la quantité disponible est insuffisante (les parties plus sombres de l'image indiquent une diminution de la concentration en eau).

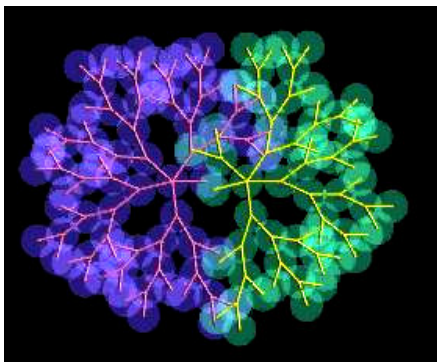


Figure 41 Compétition pour l'espace.

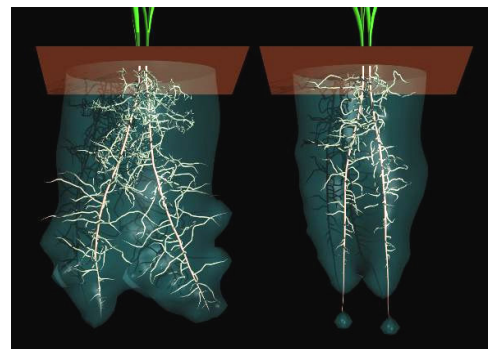


Figure 42 Influence de l'eau sur le développement de systèmes racinaires.

En utilisant les L-systèmes ouverts, il est également possible de simuler la compétition pour la lumière entre deux plantes voisines (Fig.43) ou dans un bosquet (Fig.44).

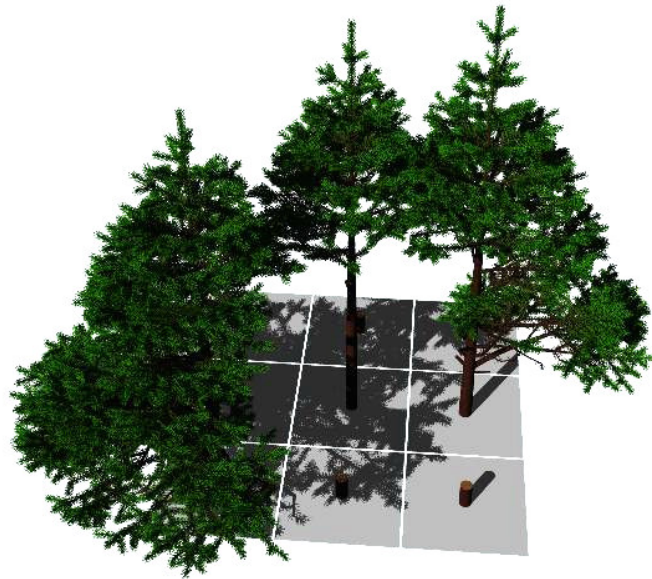


Figure 43 Compétition pour la lumière dans un bosquet.



Figure 44 Compétition pour la lumière entre arbres voisins.

4.2 DES MODELES BASES SUR LA BOTANIQUE POUR L'AGRONOMIE

« Et l'arbre entier est une pure fonction mathématique (Ceci n'est pas une affirmation; je n'ai jamais trouvé l'occasion d'une vérification.). »

Le Corbusier (Stevens, 1978)

Si la première partie de cette citation laisse à penser qu'il est aisé de modéliser et donc de simuler la croissance d'un arbre, comme une fonction mathématique, la suite, moins catégorique, rend bien compte de la perplexité que peut inspirer la structure complexe d'un végétal. Toutefois, dans le chapitre précédent (Cf.4.1), nous avons vu qu'il existe de nombreux modèles qui tentent d'appréhender la nature des plantes, modèles dont l'utilisation permet de générer, dans la plupart des cas, des plantes plus figuratives que botaniques. Notre propos ici est de définir des structures informatiques capables de représenter les caractéristiques biologiques des végétaux afin de pouvoir, d'une part, les modéliser topologiquement et géométriquement, et, d'autre part, simuler leur croissance en tenant compte des processus endogènes (croissance, ramification...) et exogènes (influence de l'environnement). Il est clair que l'on cherche à simuler des plantes virtuelles proches de la réalité biologique afin de pouvoir traiter des problèmes

agronomiques ou forestiers.

4.2.1 Modèles agronomiques et plantes virtuelles

4.2.1.1 Les différents modèles utilisés dans ce domaine

Il existe plusieurs types d'approche classique en modélisation de croissance et de production végétale :

- *Les modèles de type statistique*, basés sur les techniques de régression. Ils concernent l'évaluation du comportement moyen des peuplements et des couverts. La plante individuelle n'est évidemment pas concernée.
- *Les modèles de production végétale* en milieu contrôlé intègrent des paramètres physiques (température, lumière, ressources) et des connaissances en physiologie. Ils sont surtout utilisés sur des plantes annuelles (tomates, cotonnier). La morphologie de la plante y est ignorée, bien que la biomasse produite soit calculée.
- *Les modèles de compétition*. Chez ces derniers, des paramètres morphologiques rudimentaires sont pris en compte. Dans le cas d'applications forestières, des modèles simples de houppier (cône, ellipsoïde) sont mis en œuvre et permettent l'estimation de la production de bois en fonction de la densité. Les interactions entre arbres sont gérées par les contacts entre volumes simples engendrés.
- *Les modèles morphologiques* qui construisent des maquettes tridimensionnelles par simulation informatique. Ils sont basés sur les connaissances botaniques en architecture végétale, et sont avant tout descriptifs. L'architecture végétale possède en effet ses propres règles d'organisation interne, bien que le résultat d'une croissance soit lié aux paramètres physiques du milieu dans un environnement donné. Une même structure végétale peut avoir été réalisée plus ou moins rapidement. La plupart des modèles morphologiques existants proviennent de l'informatique et mettent en œuvre des algorithmes spécialisés dans la construction d'arborescences. Les connaissances qualitatives intégrées de ces modèles sont assez sommaires et leur utilisation se situe essentiellement dans le domaine de l'image de synthèse.
- *D'autres types de modèle spécialisés* sont utilisés en mécanique ou en transfert de matière.

En définitive, tous ces modèles se complètent les uns les autres et, de toute évidence, un modèle général qui engloberait à la fois les problèmes de morphologie et d'interaction avec le milieu aurait une polyvalence remarquable dans ses applications agronomiques. C'est cette approche que nous allons particulièrement développer.

4.2.1.2 Exploitation des maquettes tridimensionnelles

Les modèles morphologiques permettent en particulier de générer des *maquettes tridimensionnelles* des plantes simulées. Par maquette ou *plante virtuelle*, on entend « une description géométrique de la plante qui permet de la représenter dans l'espace à l'aide de polygones ».

Ces plantes virtuelles peuvent être utilisées pour de nombreuses applications et dans des domaines divers. En effet, elles permettent d'obtenir une représentation détaillée de couverts végétaux, représentation pratiquement inaccessible par des mesures *in situ*. En particulier, elles se prêtent particulièrement bien à la simulation numérique des transferts radiatifs.

4.2.2 Les logiciels de simulations AMAP

4.2.2.1 Concepts de base

Depuis plusieurs années, des équipes de botanistes et d'informaticiens essaient de modéliser et de simuler la

croissance et l'architecture des végétaux à partir des connaissances morphogénétiques acquises. Cette approche, développée au Programme de Modélisation des Plantes du CIRAD, s'appuie sur les connaissances qualitatives apportées par l'école de Hallé et Oldeman en architecture et sur les méthodes quantitatives mises au point dans le cadre de l'unité. Ces méthodes sont basées sur la description du fonctionnement des bourgeons (croissance, mort, ramification) par des processus stochastiques. Les logiciels de croissance issus de ces théories permettent de construire des maquettes réalistes de végétaux dans lesquelles la topologie (agencement des entités constituant la plante) et la géométrie (disposition de ces entités dans l'espace) sont simulées d'après les paramètres du modèle, eux-mêmes estimés à partir des données expérimentales prises sur les plantes. Ces maquettes permettent tout d'abord de visualiser en 3D l'architecture des plantes, mais peuvent également servir de base pour des applications particulières. Elles permettent par exemple de simuler le rayonnement solaire pénétrant dans un couvert végétal et de calculer les transferts radiatifs.

4.2.2.2 *Données nécessaires à la simulation*

L'analyse des mesures faites sur le terrain conduit à la constitution d'un fichier de paramètres contrôlant la croissance et le développement des plantes. Ces paramètres peuvent être regroupés comme suit :

- les paramètres morphologiques : ils définissent les caractéristiques de production inhérentes à la plante (nombre de feuilles par nœud, nombre de fleurs par inflorescence...),
- les paramètres de fonctionnement du bourgeon (loi du nombre d'entre-nœud par unité de croissance, processus de ramification...),
- les paramètres géométriques (phyllotaxie, angle de branchement, longueur des entre-nœuds...).

4.2.2.3 *Simulation de la croissance*

On peut distinguer trois notions importantes dans la simulation de la croissance des plantes :

- la topologie : elle décrit l'arrangement des entités élémentaires (entre-nœuds, feuilles, fleurs et fruits) qui constituent la plante,
- la géométrie : elle permet de positionner et de dimensionner dans l'espace ces entités,
- la dynamique de croissance : elle permet de séquencer dans le temps les différents événements intervenant lors de la croissance de la plante.

Ces notions se traduisent concrètement par une structuration des données et des programmes qui tend à suivre l'organisation de la plante elle-même.

4.2.2.4 *Différentes générations de logiciel*

Depuis sa création, le laboratoire AMAP s'est spécialisé dans le développement des modèles de croissance des plantes (encore appelés *moteur de croissance*) qui diffèrent par les connaissances botaniques qu'ils intègrent ou par les techniques informatiques qu'ils utilisent (Fig.45). Ces modèles ont donné naissance à toute une génération de logiciels (la gamme AMAP) qui correspondent à l'implémentation successive de différents paradigmes de modélisation (modèles stochastiques de la trajectoire des méristèmes, automates discrets probabilistes, modèles résistifs, automates discrets hiérarchisés, etc.). En particulier, la dernière génération intègre plus de connaissance sur la physiologie végétale (fonctionnement de la plante) et permet de simuler la plasticité de l'architecture de la plante en interaction avec son environnement (stress hydrique, lumière, compétition spatiale...). Plus récemment, des collaborations avec d'autres équipes à l'échelle internationale nous ont permis d'envisager d'intégrer dans d'autres formalismes l'idée de croissance multi-échelles (L-systèmes multi-échelles). A une échelle plus macroscopique, des modèles de croissances de peuplements, à bases dendrométriques (modèles matriciels, modèles arbres dépendants et indépendants des distances) sont élaborés pour étudier et simuler la dynamique des peuplements forestiers (régénération, croissance, mortalité) et viennent compléter la gamme AMAP.

Plus précisément, la première génération (le logiciel AMAP) est spécialisée dans la conception et la simulation de paysages virtuels (Cf.5.2.1).

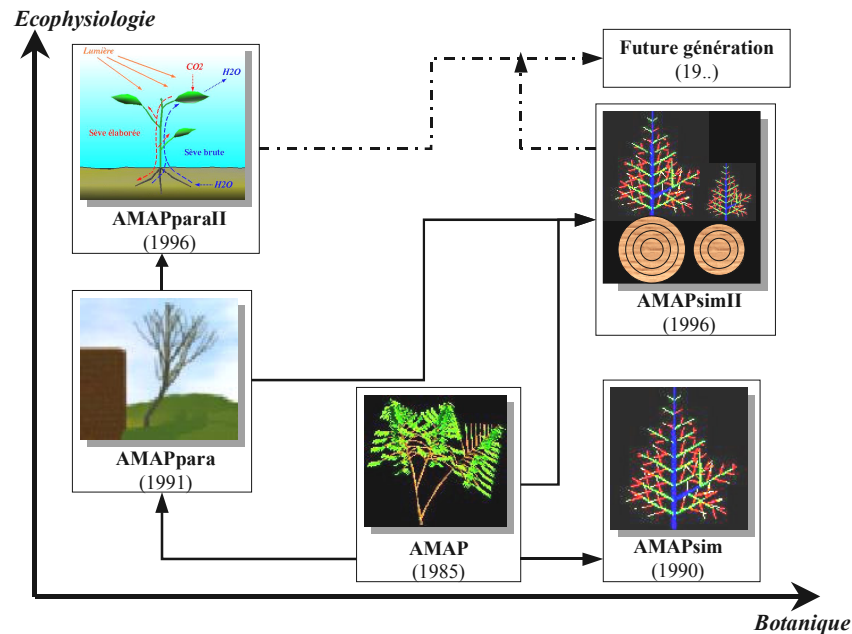


Figure 45 Les différentes générations de logiciel de simulation AMAP.

Conceptuellement, le logiciel AMAPpara représente la seconde génération des moteurs de croissance. Comme la première génération, il est basé sur des concepts botaniques très proches de la description qualitative de l'architecture des plantes. Il se base notamment sur la notion de classification des axes par ordre de ramification. Mais il diffère par son mode de gestion des événements dans le temps. En effet, la première génération utilise une gestion en préfixé alors qu'ici on gère les événements en parallèle. Cette nouvelle dynamique de croissance permet à AMAPpara, d'une part, de simuler la croissance secondaire du tronc et des branches, et, d'autre part, de tenir compte des interactions plantes/milieu (Cf.4.2.5).

La troisième génération est constituée par le moteur AMAPsim. Si sa gestion du temps est similaire à la première génération, les concepts botaniques qu'il utilise ne sont plus basés sur la notion d'ordre mais sur celle d'âge physiologique, ce qui permet, notamment, de simuler la métamorphose architecturale des plantes. De plus, l'élongation des unités de croissance et le processus de ramification sont quantifiés par des modèles mathématiques avancés (processus markoviens) qui assurent une bonne adéquation avec les phénomènes naturels. Une nouvelle version, intégrant les possibilités du logiciel AMAPpara (croissance secondaire, interactions plante/environnement), a été développée.

4.2.2.5 Des structures informatiques adaptées

Bien que ces différentes générations de logiciel aient des caractéristiques et des applications qui leur sont propres, elles doivent cependant toutes tenir compte de la réalité biologique des plantes et utiliser des structures informatiques adaptées à ce contexte. C'est pourquoi nous allons aborder dans les paragraphes suivants une description plus détaillée des modèles informatiques utilisés pour la simulation de la croissance des plantes dans le contexte AMAP. En particulier, nous exposerons les structures informatiques utilisées dans le logiciel AMAPpara. Il est clair que d'autres structures peuvent être capables de traiter une problématique botanique ou agronomique, mais l'approche AMAP a le mérite d'avoir démontré son efficacité dans de nombreuses applications, que ce soit en paysagisme, en agronomie ou en foresterie.

Nous ferons référence à quelques notions informatiques de base, notamment à celle de pointeur. Un pointeur représente une adresse mémoire permettant d'accéder à une structure ayant été allouée dynamiquement. En clair, on peut imaginer un pointeur sous la forme d'une flèche qui « relie » une structure informatique à une autre.

Reprenons donc la modélisation informatique de la croissance des plantes sous les aspects précédemment évoqués, à savoir la topologie, la géométrie et la dynamique de croissance.

4.2.3 Topologie et géométrie

Analysés par les botanistes, les végétaux laissent apparaître une structuration à plusieurs niveaux d'organisation : l'entre-nœud, l'unité de croissance (notée U.C.), la pousse annuelle et l'axe ou branche (Fig.46).



Figure 46 Structuration biologique d'un arbre.

La structure informatique mise en œuvre pour rendre compte de cette organisation est une structure hiérarchique dont chaque ordre correspond à un de ces trois niveaux. Chaque élément de plus bas niveau de cette structure définit un nœud de la plante. Les organes végétatifs (feuille, fleur ou fruit) sont également représentés par une entité spécifique. La topologie est complétée par un quatrième niveau englobant définissant la plante elle-même. Ce dernier niveau permet en fait de simuler la croissance non plus d'une seule plante isolée, mais donne la possibilité de considérer un ensemble de plantes et donc de gérer leurs éventuelles interactions. Pour chaque entité de cette structure, les liens topologiques adéquats sont créés (Fig.47) :

- La plante est constituée d'axes.
- Chaque axe est composé d'unités de croissance.
- Chaque unité de croissance compte un ou plusieurs entre-nœuds.

On utilisera les termes suivants pour spécifier la relation entre deux entités :

- On parlera d'entité *père* ou *mère* d'une autre entité si la seconde appartient à la première : un axe est *père* des U.C. et des entre-nœuds qui le composent.
- On parlera d'entité *porteuse* d'une autre entité si la première, ou un élément la composant, porte la seconde à la suite d'une ramification : un axe (sauf le tronc) a toujours un axe porteur, une U.C. porteuse et un entre-nœud porteur.

Cette structure hiérarchique permet notamment de regrouper certaines données à des niveaux particuliers en fonction de leur signification afin de limiter la redondance. Par exemple, l'ordre de ramification sera stocké au niveau de l'axe et non au niveau de chaque U.C.

L'utilisation de la structure topologique est d'ailleurs multiple. Ainsi, elle permet de stocker la géométrie de la plante au niveau le plus bas de la hiérarchie. Chaque nouvel entre-nœud apparu au cours de la croissance est défini géométriquement afin de pouvoir représenter la plante à la fin de la simulation. Ensuite, cette structure topologique permet de traiter l'élagage. En effet, dès qu'un axe est mort et élagué, toutes les entités le constituant sont supprimées de la structure topologique et elles ne seront donc plus considérées dans la suite de la croissance. Enfin, elle peut être utilisée pour mettre en œuvre des

applications particulières qui doivent toutes pouvoir accéder à la topologie de la plante traitée : taille interactive des branches, calcul de la mécanique de la plante en croissance, ou encore, intégration de mécanismes fonctionnels dans la croissance.

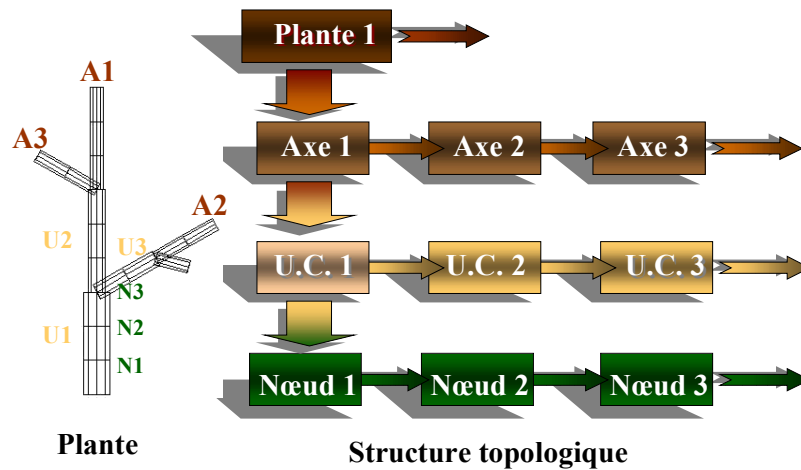


Figure 47 Structure topologique hiérarchique.

4.2.4 Dynamique de croissance

La gestion dans le temps des événements qui caractérisent la croissance de la plante peut se faire, techniquement, de différentes manières. Ainsi, une gestion simple en informatique est la gestion en préfixé (Jaeger, 1987). Mais ce type de gestion ne permet pas d'assurer la simultanéité dans le temps du fonctionnement des bourgeons actifs. L'utilisation du mode préfixé implique que la topologie de la plante simulée n'est botaniquement cohérente qu'à la fin de la simulation de la croissance. De ce fait, il n'est pas possible de faire interagir le fonctionnement de la plante avec l'environnement (climat, lumière, environnement spatial), pas plus qu'il n'est possible de simuler les échanges entre système racinaire et système aérien (sève brute, sève élaborée) ou encore de calculer l'équilibre mécanique de sa structure (Fourcaud & Lac, 1996). C'est pourquoi, une gestion en parallèle a été choisie afin de maintenir la cohérence botanique de la topologie tout au long de la croissance de la plante. La figure 48 nous montre, sur un cas très simple, les conséquences sur la structure topologique de l'utilisation des modes de gestion en préfixé et en parallèle.

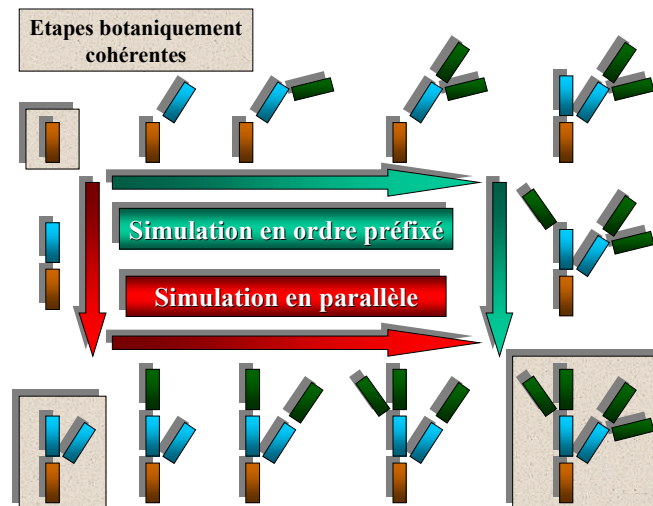


Figure 48 Simulation en préfixé et en parallèle.

La gestion en parallèle des bourgeons sur une machine séquentielle est réalisée à l'aide des techniques de la simulation à événements discrets (Blaise, 1991). Le principe général consiste à discrétiser le temps, et à définir pour chaque date une liste d'événements à traiter. Ces événements sont de différente nature : croissance des bourgeons terminaux ou des bourgeons axillaires, mise en place d'une feuille, d'une fleur ou d'un fruit.

4.2.4.1 La simulation à événements discrets : aspect théorique

La simulation à événements discrets est une technique de modélisation permettant de construire une abstraction de la réalité, et de faire évoluer cette abstraction en fonction du temps (Leroudier, 1980).

Dans une simulation à événements discrets, les variables que l'on désire connaître à tout instant sont discrètes et sont appelées variables d'état. L'ensemble des valeurs que peuvent prendre ces variables constitue l'espace d'état du système. Il s'ensuit que l'espace d'état du système est dénombrable ou fini. Du fait de la définition de l'espace d'état, on voit donc que chaque changement d'état ou événement se produit de manière discrète dans le temps à des instants précis. Ces instants de changements d'état sont appelés temps d'occurrence d'événement ou dates d'événement.

Les événements arrivant d'une manière discrète, ils peuvent être ordonnancés dans le temps (ordre chronologique) et donc simulés sur une machine séquentielle l'un après l'autre. Le parallélisme qui peut apparaître à l'utilisateur au niveau du fonctionnement logique du système, n'existe pas, en fait, au niveau des changements d'état. On parle de quasi-parallélisme, parfois aussi de pseudo-parallélisme.

Selon le mode de gestion du temps dans une simulation à événements discrets, on distingue les simulations dirigées par une horloge et les simulations dirigées par événements :

- *Simulation dirigée par une horloge*

Dans ce type de simulation, il faut définir une unité de temps appropriée au problème, et on dispose d'une horloge centrale qui progresse par pas d'une unité de temps. A chaque incrémentation de l'horloge, la liste des événements est explorée et tous les événements apparaissant à cette date sont activés.

- *Simulation dirigée par événements*

Les seuls temps accessibles lors de la simulation sont des dates d'événements et l'incrément du temps se fait d'une date à l'autre. Le temps de la simulation est alors géré par un échéancier qu'on peut présenter conceptuellement comme une liste linéaire d'événements (Fig.49). La tête de la liste est l'événement courant (présent) et la fin de la liste, l'événement le plus éloigné dans le futur. Il faut remarquer qu'à part l'événement courant, les autres événements sont purement potentiels car leur apparition peut être remise en cause lors du traitement des événements qui les précèdent.

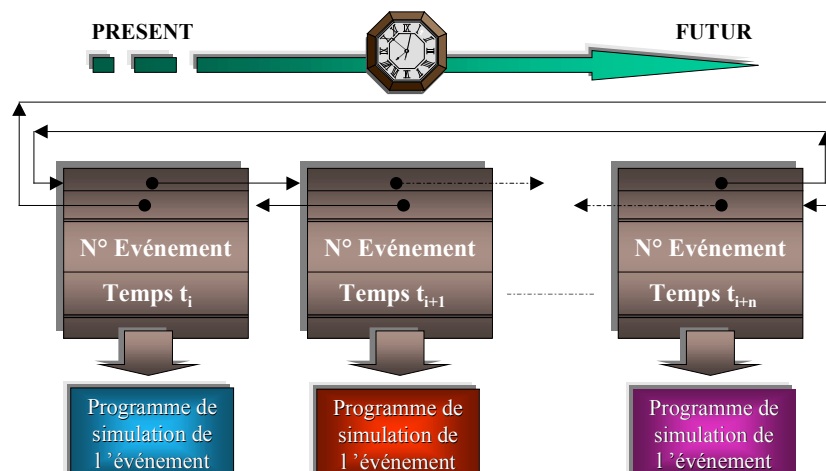


Figure 49 Liste linéaire d'événements.

D'autre part, il faut noter que lorsque deux événements sont simultanés (événements devant se dérouler au même instant), leur ordre d'apparition va dépendre de la manière dont l'échéancier est mis à jour ou balayé.

Pour pouvoir réaliser une simulation, en plus de l'échéancier, il faut des procédures qui permettent de l'entretenir et de le manipuler. L'ensemble de ces programmes et de ces structures de données s'appelle le noyau de synchronisation.

Un événement est un changement de l'état du système considéré. Lors d'une simulation, l'occurrence d'un événement va donc provoquer l'exécution d'un programme qui réalise ce changement d'état. Chaque exécution de tels programmes représente une activité de la simulation. Il existe deux approches pour réaliser ce mécanisme:

1. Le noyau de synchronisation fournit des fonctions élémentaires permettant d'ordonner les événements dans le temps. Lorsque l'événement considéré arrive en tête de l'échéancier, on exécute alors le sous-programme associé à l'événement qui décrit le changement d'état considéré dans le système (activité). On dit alors que le noyau de synchronisation (ou la simulation) est basé sur la notion d'événement.
2. Chaque activité de la simulation, qui demeure toujours la description d'un changement d'état, donc d'un événement, est vue comme l'activité d'un processus. Le modèle simulé est alors décrit comme un ensemble de processus progressant parallèlement dans le temps. Chaque processus représente une entité qui peut évoluer dans le temps au contraire des autres entités de la simulation qui restent passives. Dans une telle mise en œuvre, on dit que le noyau de synchronisation est basé sur la notion de processus.

Du point de vue du résultat, les deux approches sont équivalentes, mais elles diffèrent par leur facilité à décrire un problème donné. La première est bien adaptée à un problème posé en terme d'événements ; la seconde est mieux adaptée à un problème en terme d'activités se déroulant en parallèle. De plus, elle permet généralement de faire abstraction de l'implémentation et de rester à un niveau logique de description du système simulé.

4.2.4.2 Application de la simulation à événements discrets à la problématique plante

En tout premier lieu, nous allons éliminer la simulation dirigée par une horloge car on ne peut précisément définir dans le temps les différents états de la plante au cours de sa croissance. Nous allons donc nous intéresser exclusivement à la simulation dirigée par des événements. Il nous faut maintenant préciser les différents paramètres qui caractériseront notre système.

La simulation de la croissance des plantes nécessite de simuler l'activité de ses bourgeons. Par ailleurs, l'entre-nœud constitue l'entité de plus bas niveau dans la structure topologique (Cf. Fig.27). C'est pourquoi nous avons choisi le nœud, c'est à dire l'extrémité terminale de l'entre-nœud, comme unique variable d'état. Les différents états que peut prendre la variable d'état *nœud*, sont au nombre de trois, conformément aux différents états biologiques d'un bourgeon : il peut être actif, mort ou en pause.

Les événements qui permettent au système d'évoluer dans le temps sont au nombre de sept :

1. création d'une nouvelle unité de croissance,
2. apparition d'un nouvel entre-nœud dans une U.C. déjà commencée,
3. ramification, ce qui entraîne donc la création d'un nouvel axe et d'une nouvelle U.C.,
4. apparition d'un organe végétatif (feuille, fleur, fruit),
5. apparition d'une réitération traumatique,
6. apparition d'une réitération adaptative,
7. mort du bourgeon.

Quant à la gestion du temps, comme nous l'avons vu dans la théorie, elle se fera à l'aide d'un échéancier (Cf. Fig.49). L'unité de temps de cet échéancier ne sera pas le temps réel (la minute, l'heure ou le jour) mais une unité virtuelle dont la valeur dépendra de l'horloge interne de la plante. Nous expliciterons ce temps virtuel dans la description de la structure informatique de l'échéancier.

Il nous reste maintenant à déterminer le noyau de synchronisation qui regroupe les programmes et les structures de données nécessaires à la gestion et à la manipulation de cet échéancier. Rappelons tout d'abord le but fixé.

On cherche à simuler la croissance des plantes de façon à avoir une plante botaniquement cohérente à chaque unité de temps. Ainsi, à tout moment dans le temps, il existe un ensemble de nœuds actifs qui doivent morphologiquement évoluer. Cette évolution se traduit par l'apparition de nouveaux entre-nœuds ou organes végétatifs. La solution idéale, dans notre cas, pour simuler les changements d'état serait de décrire le modèle comme un ensemble de processus progressant parallèlement dans le temps, chaque processus ayant sa propre activité spécifique. Ainsi, à chacun de ces processus serait associé un sous-ensemble des nœuds actifs, chacun étant concerné par l'événement traité par le processus. Par exemple, au processus exécutant l'événement *création d'une nouvelle U.C.* serait associée l'entité *ensemble des nœuds devant créer une nouvelle U.C.*, et au processus exécutant l'événement *apparition d'un organe végétatif* l'entité *ensemble des nœuds devant porter un organe*. Chaque sous-ensemble serait mis à jour après l'exécution de son activité, le système évoluant ainsi jusqu'à cessation de toute activité des bourgeons de la plante. Il nous faut remarquer que la réunion de ces sous-ensembles constitue à tout moment l'ensemble des nœuds actifs, et surtout, qu'ils sont disjoints deux à deux. En effet, on peut considérer que chaque nœud ne subit qu'une seule modification morphologique à la fois. Ne disposant pas d'un langage de programmation de processus, nous nous sommes engagés dans une voie intermédiaire entre un noyau basé sur la notion d'événement et un noyau basé sur la notion de processus. Ainsi, chaque événement sera inséré dans l'échéancier en respectant l'ordre chronologique. En tête de l'échéancier se situe l'événement qui sera traité en exécutant l'activité correspondante. En fait, nous verrons, lors de la description de l'échéancier, que plusieurs types d'événements peuvent avoir lieu en même temps dans l'échéancier. Ceci traduit l'existence d'activités botaniques parallèles dans la croissance d'une plante. De plus, tous les événements de même type, situés au même moment, seront placés dans une même liste (Cf. Fig.50)

4.2.4.3 Une structure informatique pour l'échéancier

La gestion du temps dans notre simulation à événements discrets est assurée par un échéancier (Fig.50). Il se présente sous la forme d'une liste linéaire dont chaque élément représente un instant donné dans le temps décrit par l'horloge interne de la plante considérée. La liste est ordonnée de façon croissante dans le temps et sa tête représente donc l'instant présent. A chaque étape matérialisée dans l'échéancier (encore appelée *top horloge*), correspond un certain nombre d'événements à traiter. En fait, ces événements représentent l'ensemble des évolutions morphologiques que doit subir la plante à ce moment précis.

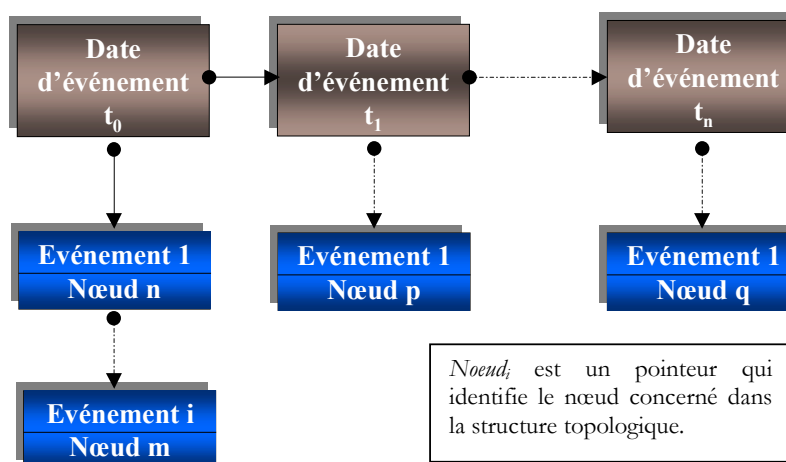


Figure 50 Structure informatique de l'échéancier.

Chaque élément de l'échéancier est constitué de quatre listes linéaires, chacune constituée d'événements de même type, et chaque événement concernant un nœud *non mort* (en pause ou en activité) de la plante :

- liste des événements *nouvelle U.C.*,
- liste des événements *nouvel entre-nœud dans une U.C. déjà commencée*,
- liste des événements *ramification*,
- liste des événements *apparition d'un organe végétatif*.

On peut remarquer que nous avons précédemment présenté sept types d'événement. Le fait que l'on n'en retienne que quatre s'explique par la similitude de certains traitements. Ainsi, le traitement des réitérations adaptatives (resp. réitérations traumatiques) est proche de celui des ramifications (resp. apparitions de nouvelles U.C.), du moins dans leur mode d'apparition au cours de la croissance. Par ailleurs, la mortalité d'un nœud n'entraîne pas la création d'un événement à insérer dans une liste, mais empêche tout simplement toute activité sur le nœud concerné.

Chaque élément des listes précédemment citées est constitué d'un pointeur pointant sur l'entité de la structure topologique qui doit subir l'événement (Fig.51). Ainsi, on a directement accès à tous les paramètres nécessaires à la caractérisation de la nouvelle forme à apparaître, entre-nœud ou organe végétatif. Cette entité est le lien entre la dynamique de croissance, puisqu'elle se trouve insérée dans une des listes de l'échéancier, et la topologie, puisqu'elle possède un pointeur sur le nœud dans la structure topologique, qui est l'objet de l'exécution de l'activité considérée.

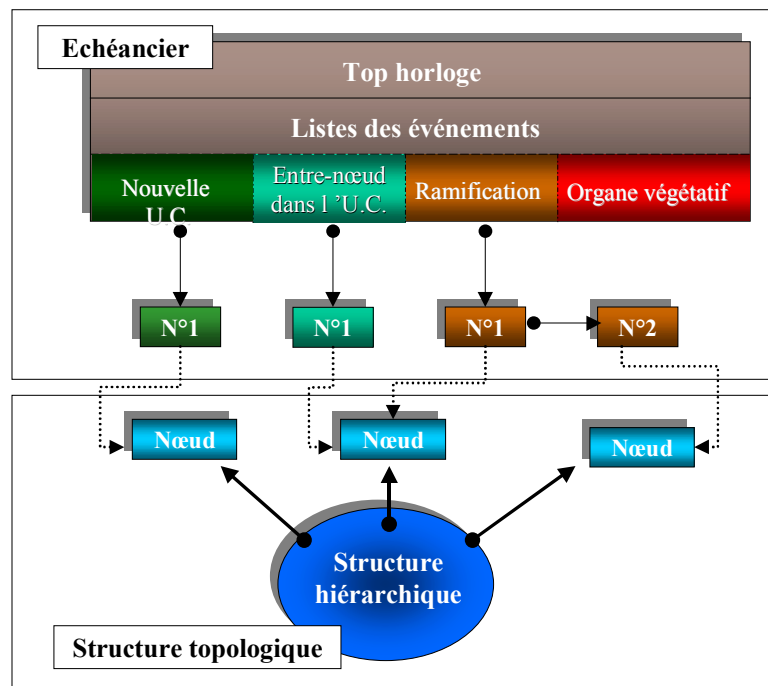


Figure 51 Liens entre l'échéancier et la structure topologique.

Notons que la notion de traitement en parallèle pour ces événements se base sur une unité de temps correspondant à un cycle de croissance. Ainsi, l'unité de temps dans cet espace de temps discrétisé correspond au délai nécessaire pour la mise en place d'une unité de croissance (U.C.) du tronc ou de la tige principale. Si l'on considère une plante se développant dans un climat tempéré et ne mettant en place qu'une seule U.C. par an, le pas de temps sera donc d'une année. Par contre, si l'on considère une plante comme le cotonnier, le pas de temps pourra varier de 2 à 10 jours en fonction des conditions climatiques.

Par ailleurs, il est important de signaler que les dates d'apparition des événements caractérisant la croissance de la plante peuvent ne pas être des multiples du pas de temps. Par exemple, des axes peuvent avoir une croissance plus rapide que leur axe porteur, ou encore, la simulation de mécanismes fins peut nécessiter la prise en compte d'événements au cours de la durée du cycle de croissance. Aussi, notre

espace de temps est discret car composé de valeurs distinctes, mais on peut passer d'une valeur à une autre par une quantité non entière. En d'autres termes, une date d'événement est un réel dans le domaine fini, car dénombrable, du temps discrétisé. Ainsi, des événements peuvent s'insérer à des dates intermédiaires si leur vitesse de fonctionnement l'exige.

4.2.4.4 Mise à jour de l'échéancier

Une procédure du noyau de synchronisation s'occupe de l'entretien de l'échéancier. Cette procédure se charge d'insérer au top horloge adéquat les événements à traiter.

Chaque fois qu'un nouvel entre-nœud ou organe végétatif apparaît lors de la croissance de la plante, il est tout d'abord placé dans la structure topologique, puis une variable pointant sur cette nouvelle entité est définie. C'est à partir de cette variable que vont être créés tous les événements que subira éventuellement cette nouvelle entité, événements qui seront ensuite insérés dans l'échéancier à la date et dans les listes convenables, et ceci en fonction de leur nature. En fait, la procédure de mise à jour de l'échéancier ne sera appelée que lorsqu'un nouvel entre-nœud sera créé, et non pas un organe végétatif fixe (feuille, fleur, fruit). En effet, ceux-ci sont *statiques* dans le temps (ils ne sont pas constitués de bourgeons actifs) et ne peuvent donc générer de nouveaux événements dans l'échéancier. Ainsi, l'évolution morphologique du nouveau nœud peut entraîner l'activation éventuelle de plusieurs événements, ceux-ci n'étant pas forcément insérés à la même date dans l'échéancier. Considérons un nouveau nœud :

- Il peut continuer sa croissance, et par conséquent, soit donner naissance à une nouvelle U.C., soit prolonger l'U.C. à laquelle il appartient.
- Il peut également porter des organes végétatifs.
- On peut voir apparaître une ramification.
- Enfin, un système réitéré adaptatif peut voir le jour.

Une fois la date dans l'échéancier définie, il reste encore à insérer un nouvel événement dans la liste des événements de même type. Ainsi, il a été décidé dans un premier temps de trier les listes par valeur croissante de l'ordre de ramification des axes auxquels appartiennent les nœuds concernés par les événements. Ceci s'explique par le fait que les axes d'ordre inférieur sont les axes les plus importants dans une plante, et qu'il peut être intéressant dans certains cas de les traiter avant les axes de moindre importance. Par contre, l'insertion d'un événement parmi d'autres concernant des nœuds dont l'axe père est de même ordre, se fait de façon aléatoire. Ceci permet d'éviter que des axes soient toujours traités avant d'autres axes de même ordre de ramification. Un tel biais systématique pourrait entraîner des artefacts dans la simulation.

Enfin, notons que tous ces événements, bien qu'étant insérés dans l'échéancier, ne sont pour autant pas sûrs d'aboutir. En effet, ce sont les activités qu'ils déclencheront, qui décideront au vue d'un certain nombre de tests si l'événement a effectivement pu se produire.

4.2.5 Interactions plante/environnement

La croissance des plantes reflète l'influence des facteurs climatiques et biologiques qui, en influençant des processus physiologiques internes, contribuent peu ou prou à la formation et au fonctionnement des bourgeons. Comme la croissance est influencée par un ensemble de conditions écologiques, il est très difficile d'apprécier la part d'un facteur spécifique de l'environnement dans la croissance de la plante, car ces conditions sont interactives et interdépendantes (Kozłowski, 1949). Parmi les composantes principales de l'environnement influençant la croissance, on trouve la lumière, l'eau, la température, l'apport des sels minéraux, la composition de l'atmosphère (le CO₂ en particulier), les propriétés physiques et chimiques du sol, les insectes, les autres plantes et de nombreux animaux. L'influence des facteurs génétiques et de l'âge de la plante se superpose à celle de l'ensemble des facteurs de l'environnement.

Nous allons nous intéresser plus particulièrement à deux de ces facteurs : les autres plantes et la lumière.

En fait, lorsque l'on parle des autres plantes, on fait allusion au phénomène de gêne aussi bien entre

plantes voisines qu'à l'intérieur même d'une plante donnée. Le problème des obstacles fixes (les murs par exemple) sera également abordé.

Quant à la lumière, sous ce terme générique se cachent plusieurs aspects. En effet, la croissance des plantes est influencée par l'intensité de la lumière, par le temps d'exposition à la lumière (photopériode) et par la qualité (longueur d'onde) de la lumière. Mais il semblerait que, dans des conditions naturelles, l'intensité de la lumière et la photopériode aient une influence plus grande que sa qualité (Kozlowski, 1949). Nous nous intéresserons à quantifier la lumière reçue par la plante et à représenter la recherche de la lumière par certaines plantes au cours de leur croissance.

Mais avant de pouvoir traiter ces différents aspects, il nous faudra aborder dans un premier temps un problème plus technique : la définition de l'espace dans lequel évolue la plante.

4.2.5.1 Discretisation de l'espace et des plantes

L'un des problèmes qui se présentent à nous réside dans la résolution des collisions entre entités géométriquement connues. La solution la plus évidente est d'utiliser la géométrie analytique qui, à l'aide d'outils classiques, nous donnera entière satisfaction dans la plupart des cas. Si pour des volumes ou des surfaces simples, et en nombre limité, le coût en complexité et en temps calcul semble supportable, il nous est apparu intéressant de trouver une solution adaptée au cas des plantes, pour lesquelles, si les formes géométriques à manipuler sont simples, le nombre d'éléments à traiter est toujours très important. Aussi avons-nous utilisé la technique de l'espace *voxel*. Un espace *voxel* est défini comme étant une région de l'espace 3D discrétisé en cubes ou *voxels* de même dimensions. Cette méthode a donné de bons résultats tant dans l'imagerie médicale pour visualiser les données 3D produites par des appareils spécialisés, qu'en image de synthèse pour créer des surfaces définies par des fonctions. Par ailleurs, cette méthode a été utilisée dans un cadre qui nous intéresse tout particulièrement : les interactions entre une plante et son environnement (obstacle, ombrage) lors de sa croissance (Fig.52) (Green, 1989).

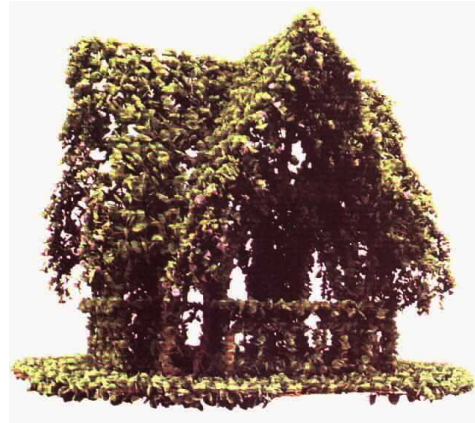


Figure 52 Lierre recouvrant une maison.

Grâce à cet espace discrétisé, il est plus facile et plus rapide de déterminer certaines relations, telles la proximité ou l'intersection, entre des objets géométriques, que par l'utilisation de la géométrie analytique. En effet, la seule opération géométrique nécessaire dans un espace discrétisé consiste à pouvoir insérer dans cet espace les objets à traiter, dans notre cas des plantes et d'éventuels obstacles fixes, toutes les autres opérations se résumant à une comparaison du contenu des cubes. Mais encore faut-il définir le contenu d'un voxel, car la complexité de cette opération de comparaison en dépend directement.

Le voxel est donc un cube qui contient une description d'une partie de l'espace délimitée par ses dimensions. Suivant la valeur de la dimension d'un voxel, la discrétisation de l'espace sera plus ou moins fine. Intuitivement, on peut penser que le choix de cette dimension, que l'on appellera *pas de discrétisation*, influe sur les résultats que l'on peut obtenir. En effet, suivant le pas de discrétisation et la taille des objets considérés, plusieurs voxels peuvent être occupés par un même objet, et des individus distincts peuvent occuper des parties différentes d'un même voxel sans pour autant entrer en contact. Dans de nombreuses applications, un voxel est de nature binaire, c'est à dire qu'il spécifie, par une valeur logique, l'absence ou la présence d'un objet dans l'espace qu'il délimite. Dans ce cas, la structure du voxel ne permet pas de gérer la présence d'objets différents dans un même voxel, ce qui sous-entend que le pas de discrétisation doit être défini en conséquence. De plus, la valeur du voxel ne permettant pas d'identifier l'objet l'occupant, on ne peut effectuer aucun calcul géométrique. Par contre, la structure du voxel en est particulièrement simplifiée. Si l'on souhaite pouvoir traiter des voxels pouvant contenir en partie plusieurs objets, il faudra leur assurer une structure suffisante qui leur permette de stocker différents individus, mais aussi de les

reconnaître. La solution évidente est de représenter chaque voxel par une liste de nombres entiers, chacun identifiant un objet particulier. Dans tous les cas, si l'on veut calculer et gérer des intersections, le passage à la géométrie analytique est obligatoire. Cette approche implique que le voxel puisse donner accès à la connaissance géométrique des objets. Pour ce faire, il suffit qu'à chaque identifiant corresponde la description géométrique de l'objet. Bien qu'utilisant la géométrie analytique, on limite le coût en temps calcul en restreignant l'espace considéré à un unique voxel.

Dans le cas qui nous intéresse, une plante réagira différemment suivant la nature de l'objet responsable de la gêne (mur, axes de la même plante ou d'une plante différente). De plus, dans le cas d'une plante, lors d'une ramification, l'axe porteur voit automatiquement sa première U.C. appartenir, au moins en partie, au voxel déjà occupé par l'U.C. de son axe porteur qui a ramifié. Il nous faut donc pouvoir associer à chaque voxel les informations suffisantes afin de traiter la présence de plusieurs individus dans un même voxel, et pour être capable de les identifier.

Si nous avons maintenant approché le problème de la définition du voxel, sa description reste encore insuffisante. En fait, nous préciserons sa structure au fur et à mesure de la présentation des applications qui l'utilisent. En effet, c'est la méthode élaborée pour la résolution de la gêne intra et inter couronne qui fixera définitivement sa nature.

4.2.5.2 La plante dans un espace voxel

La structure informatique de l'espace voxel peut prendre des formes différentes. Une première approche consiste à représenter l'espace voxel par une structure permettant de différencier les zones vides des zones pleines ou partiellement occupées. Cette méthode a l'avantage de minimiser la place mémoire occupée, les parties non utilisées de l'espace n'étant pas décrites. Par contre, la mise en place et la gestion informatique d'une telle structure est assez complexe. La seconde représentation possible de cet espace est plus simple, mais ne tient pas compte du coût mémoire. Il s'agit d'utiliser un tableau à trois dimensions (X, Y et Z) dont chaque entrée définira la structure d'un voxel. Que l'entrée soit vide ou non, la place mémoire sera réservée. Dans la suite de l'exposé, nous considérerons que c'est cette deuxième solution qui a été choisie afin de simplifier l'exposé : elle privilégie la simplicité de la gestion de la structure et la rapidité de l'accès aux données, plutôt que la limitation du coût mémoire. Pour plus de précisions sur les différentes méthodes de subdivision de l'espace, et notamment pour une comparaison du coût des traitements suivant l'option choisie, on pourra consulter les travaux de Devillers (1990).

La plante est constituée d'entre-nœuds et d'organes végétatifs (feuilles, fleurs et fruits). Chacun de ces éléments est présent dans la structure topologique et est parfaitement défini géométriquement : directions principale et secondaire, diamètres initial et final, coordonnées de la base et de l'extrémité de l'axe principal. Les positionner dans l'espace voxel (Fig.53) nécessite leur discrétisation, c'est à dire le calcul de l'ensemble des voxels qu'ils occupent. Nous allons plus particulièrement nous intéresser aux entre-nœuds.

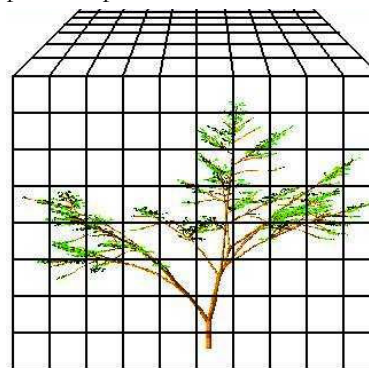


Figure 53 Plante dans un espace voxel.

4.2.5.3 Méthode de discrétisation des entre-nœuds

Un entre-nœud est une forme volumique définie en 3D, mais, pour simplifier le problème, considérons d'abord qu'il est uniquement représenté par son axe principal. Autrement dit, nous allons en premier lieu traiter le problème de la discrétisation d'un segment de droite. Ensuite, nous aborderons les objets volumiques. Cette décomposition est faite afin de mettre en évidence le traitement des segments de droite, car cette méthode sera utilisée dans d'autres buts que celui de discrétiser des entités définies géométriquement.

La discrétisation de segments est un problème très connu en informatique graphique surtout dans un

espace 2D : affichage sur écran ou tracé sur imprimante. Un algorithme très utilisé, car optimisé et efficace, est celui de Bresenham (Bresenham,1965). Pourquoi s'intéresser dans notre cas à ces algorithmes d'affichage ? En fait, notre problème de discrétisation est identique à celui de l'affichage d'un segment sur un écran raster. En effet, les points de l'écran (les pixels) sont les équivalents 2D de nos voxels, et l'algorithme de Bresenham, qui permet de déterminer les points à afficher, pourrait nous permettre de localiser les voxels par lesquels passe le segment.

Cependant, cet algorithme ne nous permet pas, tel quel, de résoudre complètement notre problème, et ce, pour deux raisons :

- Tout d'abord, il travaille dans un espace 2D alors que nous évoluons en 3D (même dans le cas des segments de droite). Ceci dit, le passage en 3D devrait s'effectuer en conservant la même théorie.
- Le deuxième point est plus fondamental. Comme nous l'avons dit précédemment, ces algorithmes de tracé sont approximatifs, on ne pourra obtenir par un de ces algorithmes la liste exhaustive des voxels traversés par un segment de droite (Fig.54). Si on peut comprendre que cette approximation soit suffisante à l'échelle du pixel, on ne peut en faire autant à l'échelle du voxel.

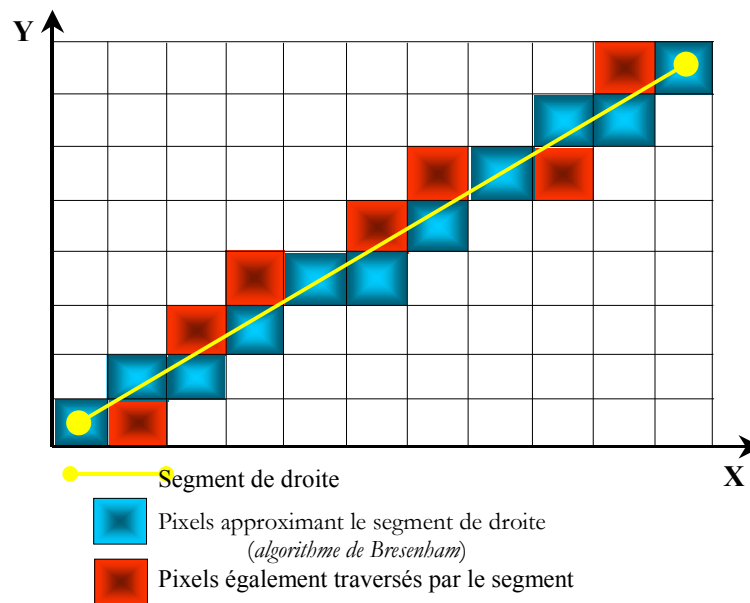


Figure 54 Discretisation d'un segment de droite.

Nous avons donc développé un algorithme qui s'inspire de la méthode de Bresenham (algorithme incrémental et généralisation d'une partie de l'espace à son ensemble) mais qui répond à nos besoins spécifiques :

- traitement des segments 3D,
- et surtout, détection de tous les voxels traversés par le segment sans exception.

Ainsi, si l'on se restreint à la partie de l'espace définie par l'équation ci-dessous, son principe général est le suivant.

$$\begin{cases} x \geq 0 \\ y \geq 0 \\ z \geq 0 \\ x > y \\ x > z \end{cases}$$

On détermine à partir des coordonnées 3D de l'extrémité initiale du segment de droite les coordonnées du voxel le contenant. Soit $P_0 = O + \lambda \vec{v}$ le point d'entrée du segment dans ce voxel. On peut considérer les voxels comme étant définis par les intersections de plans parallèles aux plans principaux xOy, xOz, yOz , et séparés d'une distance égale au pas de discrétisation. On calcule alors les intersections du segment et de chacun des plans délimitant les parties *supérieures* du voxel (Fig.55). On cherche à calculer les paramètres $\delta_x, \delta_y, \delta_z$ où δ_i ($i \in \{x, y, z\}$) est le plus petit réel positif tel que :

$$P_i = O_i + (\lambda + \delta_i) \vec{v} \quad i \in \{x, y, z\}$$

On détermine le plus petit des nombres δ_i et on considère le point P_i correspondant comme le point d'entrée du segment dans le voxel suivant. Ce point est le nouveau point initial. On réitère tant que l'extrémité finale du segment n'est pas atteinte.

Nous pouvons maintenant discrétiser une plante en filaire. Le problème du volume peut être résolu de façon assez simple à partir de cette première approche. En effet, nous connaissons la géométrie des entre-nœuds, et notamment, leurs diamètres initial et final. Ainsi, lors du calcul de chaque nouvelle intersection, on détermine, dans le plan perpendiculaire à la direction principale de l'entre-nœud, les voxels situés à une distance inférieure ou égale au rayon de l'entre-nœud à cet endroit (Fig.56).

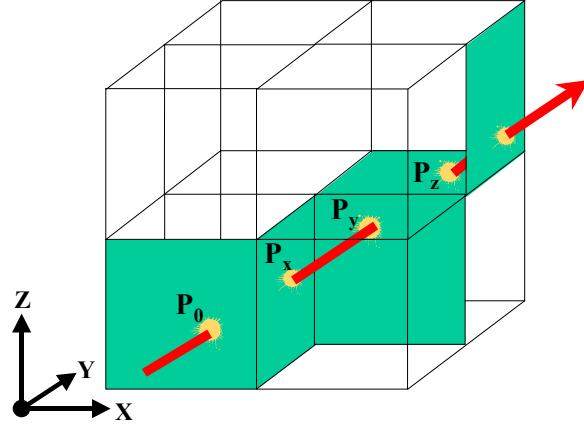


Figure 55 Discretisation d'un segment de droite.

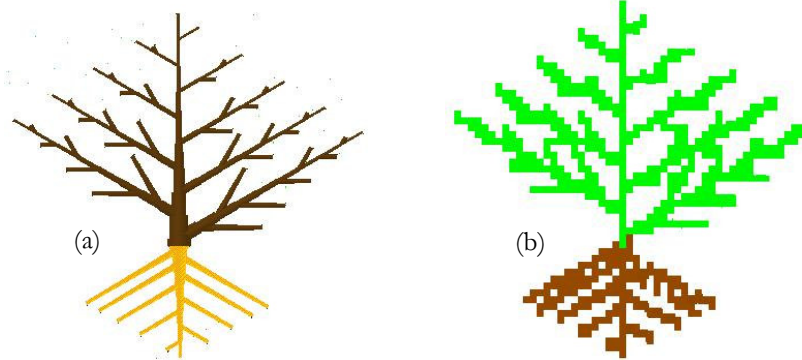


Figure 56 Voxélisation d'une plante : (a) plante volumique, (b) enveloppe de voxels.

Il nous reste maintenant à définir le contenu des voxels traversés. C'est ce que nous allons aborder en traitant le cas concret de la détection des collisions

4.2.5.4 Traitement de la compétition pour l'espace

Lors de sa croissance, une plante voit ses branches se déployer et occuper un espace toujours plus grand. Si le développement de ces branches entraîne des collisions entre certaines, on parlera de gêne. En fait, on peut qualifier cette gêne de mécanique ou de géométrique, ou encore de contact, car elle n'apparaît que lorsque l'espace nécessaire au bon développement d'un bourgeon se trouve occupé. Lorsque l'on parle de contacts entre branches, ces dernières peuvent appartenir à la même plante ou, tout au contraire, être issues de plantes différentes. On pourra donc discerner la gêne *intra-couronne* et la gêne *entre couronnes*.

Enfin, la gêne peut également être la conséquence de la présence d'un obstacle fixe (n'évoluant pas dans le temps), tel un mur, dans l'espace de développement de la plante.

Nous avons déjà évoqué le problème de la détection et de la résolution des collisions dans un espace discrétisé (Cf.4.2.5.1). En particulier, nous avons déjà en partie défini la structure et surtout le contenu des voxels. Ainsi, chaque voxel contient une liste d'identifiants, chacun permettant de reconnaître l'objet qui l'occupe. En particulier, chaque voxel doit donc être structuré de manière à pouvoir décrire les entre-nœuds qu'il contient, du moins en partie. Toutefois, dans la structure hiérarchique représentant la topologie de la plante (Cf.4.2.3), aucun identifiant n'est affecté aux entre-nœuds, seule les axes sont identifiés. Et de fait, les voxels de notre espace discrétisé contiennent les identifiants des axes pères des entre-nœuds les occupant.

Une autre difficulté apparaît alors pour détecter les collisions. Cette détection se ramène donc à comparer un nouvel entre-nœud que l'on souhaite disposer dans l'espace voxel, à des numéros d'axe. Ces numéros nous donnent accès, par l'intermédiaire de la structure topologique, à la liste des entre-nœuds qui les constituent. On peut alors comparer la géométrie du nouvel entre-nœud à celle des entre-nœuds de l'axe identifié dans le voxel. Mais cette méthode n'est pas optimale, dans la mesure où on est obligé de considérer tous les entre-nœuds formant les axes, car on ne sait pas quels sont ceux qui sont effectivement présents dans le voxel considéré. Il aurait été alors plus judicieux d'affecter un numéro d'identification à chaque entre-nœud et de stocker celui-ci dans les voxels à la place de celui des axes.

En fait, afin de simplifier et d'accélérer les traitements, nous n'avons pas choisi de détecter géométriquement les collisions, mais d'utiliser la topologie des axes dans la plante pour se faire. Toujours en utilisant la structure topologique, on accède facilement à un certain nombre d'informations concernant l'entre-nœud que l'on souhaite placer dans l'espace voxel. En particulier, les caractéristiques suivantes sont accessibles :

- numéro de l'axe père,
- ordre de ramification de cet axe,
- numéro de l'axe porteur,
- ordre de ramification de l'axe porteur,
- numéro de la plante à laquelle il appartient.

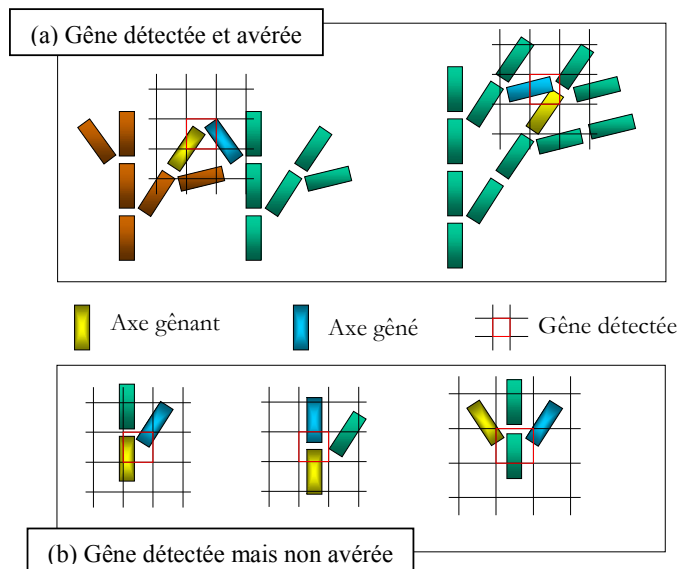


Figure 57 Détection des cas de gêne.

Pour les axes localisés dans le voxel considéré, on accède aux mêmes données, et notamment celles qui

concernent leur axe porteur. En comparant ces différentes données, on décide s'il y a gêne ou pas. Ainsi, il y a gêne si l'une des conditions suivantes est vérifiée (Fig.57.a) :

1. L'axe père et l'axe gérant n'appartiennent pas à la même plante.
2. L'ordre de ramification de l'axe gérant est inférieur (une branche ne peut être gérée par une branche plus petite) ou égal (la première branche mise en place occupe l'espace) à celui de l'axe père.

Toutefois, l'efficacité de l'espace voxel peut être mise en défaut dans certains cas (Fig.57.b). C'est pourquoi il nous faut définir d'autres critères afin de préciser la notion de gêne. Ainsi, si l'une des conditions suivantes est vérifiée, il n'y a pas de gêne :

- L'axe gérant est l'axe porteur (mise en place d'une ramification).
- L'axe gérant est l'axe père (croissance d'un axe).
- L'axe gérant et l'axe géré ont le même axe porteur (verticille).

Une fois l'éventuelle collision détectée, et le cas de gêne démontré, il reste encore à effectuer un traitement adéquat. Le traitement appliqué est simple. En effet, on ne cherche pas à déterminer de façon exacte les intersections entre les entre-nœuds. L'existence ou l'absence d'une collision sera synonyme d'inhibition ou de continuation de la croissance. Autrement dit, en cas de gêne, le bourgeon meurt, et son axe père cesse son développement (Fig.58). Il faudra attendre la prise en compte de la lumière pour voir la réaction à la gêne traitée de façon plus souple (Cf.4.2.5.7).

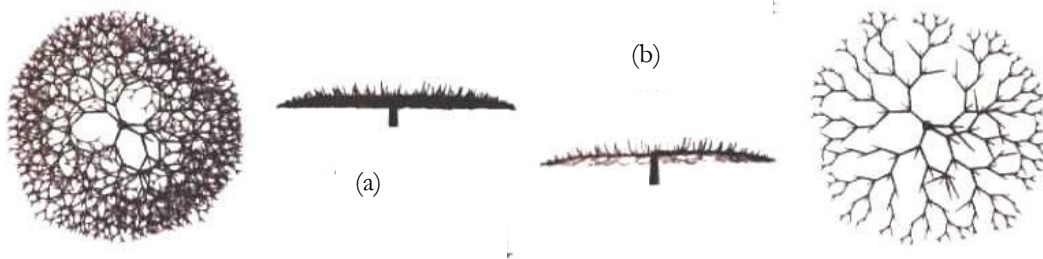


Figure 58 Compétition pour l'espace dans un étage de *Terminalia* (Blaise, 1991) :
(a) sans traitement de la gêne, (b) avec traitement.

4.2.5.5 Mise en place d'obstacles dans l'espace voxel

L'exemple courant d'obstacles fixes utilisés dans nos essais consiste en murs que l'on représente par des parallélogrammes de dimensions variables (Fig.59). Ces murs sont disposés dans l'espace discrétisé avant de simuler la croissance de la plante. La définition des obstacles est faite par l'utilisateur à l'aide du même éditeur utilisé pour les paramètres de définition de la plante. Chaque obstacle est défini par :

- ses dimensions : coordonnées réelles de son origine et du point diagonalement opposé,
- sa composition : homogène ou non, et dans ce dernier cas, on précise un facteur de densité (Fig.59),
- son matériau : opaque ou transparent. Cette différence pourra être utile lors de la simulation de la lumière.

La disposition d'un obstacle dans l'espace discrétisé se fait en initialisant les voxels qui le composent par une valeur qui permet, tout d'abord, de reconnaître qu'il s'agit bien d'un obstacle fixe, et ensuite, de déterminer sa composition (homogène ou non, opaque ou transparent). Dans le cas d'un obstacle hétérogène, on décide si un voxel, situé normalement dans les limites de l'obstacle, est occupé ou non, en comparant un nombre généré aléatoirement au facteur de densité de l'obstacle. Dans le cas d'un obstacle fixe, qu'il soit transparent ou opaque, la gêne est totale : aucun axe ne peut cohabiter avec un obstacle dans le même voxel.

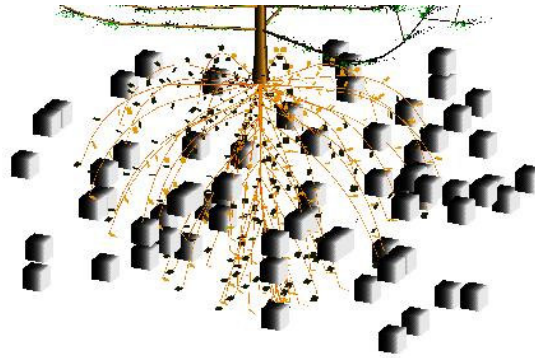


Figure 59 Exemples de mise en place d'obstacles dans l'espace voxel.

4.2.5.6 Discretisation de la voûte céleste

L'intensité de la lumière à laquelle sont exposés les plantes varie grandement suivant l'habitat, l'âge des plantes et leur position dans le cas d'un peuplement. Elle varie également beaucoup selon la partie de la couronne considérée. Par exemple, l'intensité de la lumière à l'intérieur de la couronne d'un cyprès oscille entre 0.2 et 5 % de la lumière solaire totale. Ainsi, suivant sa position dans la couronne, un bourgeon captera plus ou moins de lumière, et selon sa sensibilité, réagira en conséquence. Pour calculer la quantité de lumière reçue par un bourgeon, nous avons choisi de discrétiser l'espace en direction privilégiée, et d'estimer pour chacune la lumière incidente.

Le problème du choix de directions privilégiées dans un ensemble infini est courant dans le domaine de la télédétection, et dans toutes les applications de simulation des transferts radiatifs. Ainsi, nous nous sommes penché sur un modèle particulier qui correspond tout à fait à nos besoins: le modèle **TURTLE** (The Universal Reflection and Transmission model for Layered crop Experiment).

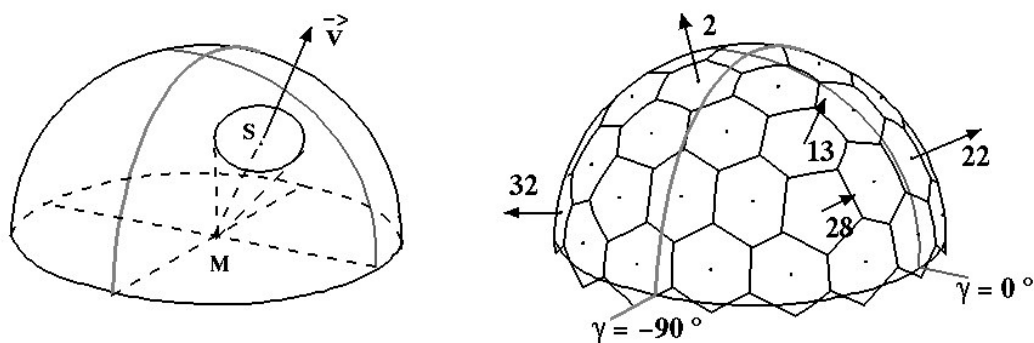


Figure 60 Le modèle TURTLE : (a) zone associée à une direction, (b) discrétisation de l'hémisphère.

Comme dans d'autres modèles numériques de réflexion de cultures, la continuité de toutes les directions possibles est remplacée par un nombre de directions de référence (Dulk (den), 1989). Chaque direction de référence représente toutes les directions dans un cône autour d'elle (Fig.60.a). Une méthode couramment utilisée pour distribuer toutes les directions dans une hémisphère à travers un ensemble de vecteurs représentatifs, consiste à diviser une surface sphérique en régions. Tous les vecteurs radiants pointant dans une région sont représentés par le vecteur radiant du centre de cette région. Les modèles qui sont basés sur ce principe s'appuient généralement sur un partitionnement suivant les méridiens et les parallèles d'une sphère. Les intersections de méridiens et de parallèles prédéfinis sont utilisées comme directions de référence. Cette méthode a cependant un désavantage : avec un intervalle constant entre les parallèles, les angles solides représentés par chaque vecteur décroissent avec l'augmentation de l'inclinaison, ainsi que les angles entre deux directions azimutales adjacentes. Quand on essaie de garder les angles solides

représentés inchangés, les cônes proches du zénith deviennent comparables à des rectangles, voir même à des triangles. Un partitionnement correct d'une surface sphérique devrait vérifier les principes suivants :

- Les angles entre les vecteurs représentatifs de régions adjacentes sont égaux.
- La forme d'une région est circulaire (Fig.60.a) afin de minimiser l'angle maximum entre les vecteurs représentatifs et les vecteurs représentés.
- Toutes les régions sont de taille égale.

Le modèle TURTLE est ainsi basé sur un partitionnement de l'hémisphère en hexagones et pentagones : 46 directions sont définies, chacune distante d'une autre adjacente d'un angle d'environ 24° (Fig.60.b).

4.2.5.7 Influence de la lumière

Nous cherchons à calculer la quantité de lumière captée par un bourgeon. Rappelons que ce bourgeon est en fait l'extrémité d'un entre-nœud que l'on cherche à placer dans l'espace voxel. L'entre-nœud mis en place, il nous faut calculer la quantité de lumière qui arrive dans le voxel contenant l'entre-nœud et son bourgeon terminal.

Nous avons précédemment défini des directions privilégiées pour la lumière. Ces directions sont connues géométriquement, on peut donc représenter les segments de droite partant du voxel contenant le bourgeon considéré et dont la direction correspond à un des vecteurs définis par le modèle TURTLE (Fig.61). Chaque segment est limité par les dimensions de l'espace voxel. Ces segments de droite sont alors discrétisés à l'aide de la méthode précédemment présentée (Cf.4.2.5.3). Pour chaque direction, on a donc la liste des voxels qu'elle traverse. A chaque direction, on associe un paramètre qui contient la quantité de lumière provenant de cette direction. Le principe général pour le calcul de la quantité de lumière est alors le suivant :

- Dans chaque direction, le bourgeon reçoit initialement la quantité maximale de lumière émise par la source lumineuse (le soleil en général).
- Pour chaque voxel qui discrétise une direction donnée, on examine son contenu et on en déduit la quantité de lumière qu'il laisse filtrer. Un obstacle opaque empêche toute lumière d'atteindre le bourgeon, alors qu'un obstacle transparent laisse filtrer toute la lumière reçue. Si le voxel est occupé par une branche, la quantité de lumière reçue dépend du pourcentage d'occupation du voxel par la branche.
- De proche en proche, on calcule ainsi la quantité de lumière reçue par le bourgeon dans une direction donnée. Une fois la quantité de lumière calculée pour chaque direction, on fait la moyenne de ces valeurs pour avoir la quantité de lumière effectivement captée par le bourgeon.

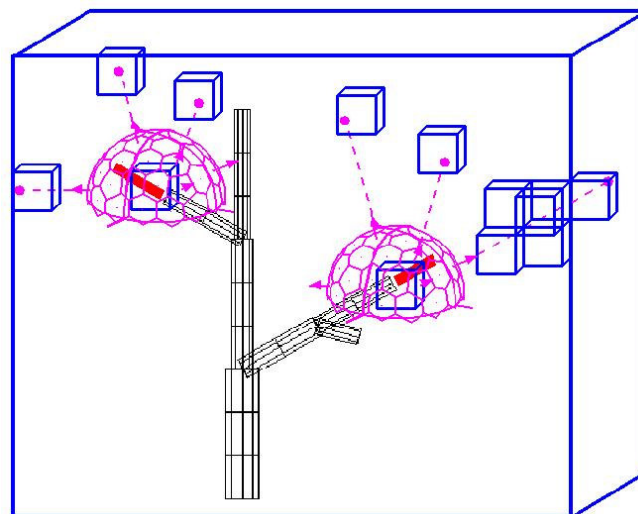


Figure 61 Calcul de la quantité de lumière reçue par un bourgeon.

Notons qu'initialement, chaque direction apporte la même quantité de lumière : on considère que la lumière est homogène dans l'hémisphère, ce qui correspond au rayonnement diffus. De même, on fait la moyenne des différentes directions, car on suppose qu'aucune n'est privilégiée par rapport à une autre.

Considérons une plante simple en 2D et un obstacle placé perpendiculairement à la plante. On active différents paramètres afin que la plante soit sensible à la gêne de contact et au manque de lumière. Dès qu'un obstacle est rencontré ou que le taux de lumière est insuffisant, la plante peut réagir de façons différentes :

- L'entre-nœud concerné meurt entraînant la mort de son axe père.
- Il cherche à dévier sa trajectoire afin de trouver des voxels moins occupés ou recevant plus de lumière. Toutefois, la déviation est limitée à un angle maximal fixé en fonction de l'ordre de ramification de l'axe père.

La figure 62 résulte d'un traitement qui diffère de celui précédemment présenté : en cas de déficience de la lumière, le bourgeon ne meurt pas mais tente de modifier sa trajectoire de croissance afin de bénéficier d'un meilleur ensoleillement. Ceci explique la déformation de la couronne de l'arbre due à l'ombre que lui porte le mur opaque. Si nous savons calculer le taux de lumière transmis dans chacune des directions du modèle TURTLE, il nous faut encore déterminer en fonction de ces taux de lumière la nouvelle direction de l'entre-nœud. On la calcule comme étant la moyenne de chaque direction pondérée par la quantité de lumière transmise (ce taux est normalisé, c'est à dire compris entre 0.0 et 1.0) par la direction considérée. Plus la quantité de lumière provenant d'une direction est importante, et plus l'entre-nœud aura tendance à se développer dans cette direction. De plus, pour éviter que le changement de direction de l'axe soit trop brutal, la moyenne des directions privilégiées est également pondérée par le sinus de l'angle entre la direction initiale de l'entre-nœud et la direction privilégiée considérée.

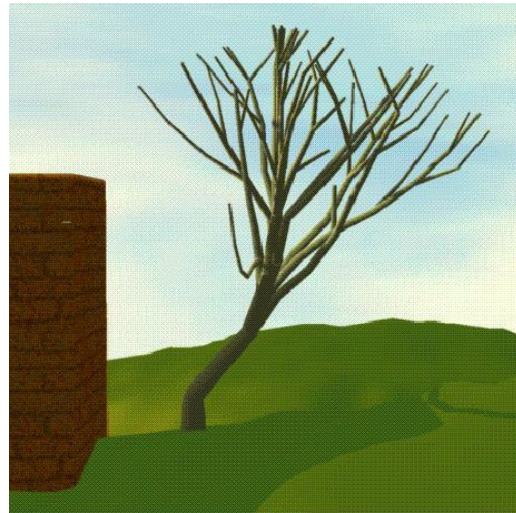


Figure 62 Recherche de la lumière
(Blaise, 1991).

4.2.5.8 Conclusion

L'emploi de la technique des voxels permet de discrétiser de façon simple l'espace 3D. On peut représenter des formes en trois dimensions, et notamment les plantes, en marquant les voxels occupés. Mais cette représentation est nécessairement approximative car, bien que pouvant identifier les objets (dans notre cas les entre-nœuds) et connaissant leur description géométrique, on ne peut connaître exactement la surface des objets occupant un voxel. Par ailleurs, discrétiser l'espace par des voxels implique une approximation dans les calculs géométriques, et impose un pas de discrétisation minimal en dessous duquel les objets ne peuvent plus être distingués. Cette méthode nécessite également de pouvoir disposer de beaucoup d'espace mémoire, espace d'autant plus étendu que les dimensions de l'espace voxel sont grandes, le pas de discrétisation petit et la structure d'un voxel complexe.

Toutefois, pour la plupart des applications, la simplicité et la rapidité des traitements équilibrent amplement ces inconvénients pour la précision acceptée. Ainsi, la figure 64 nous montre les possibilités offertes par l'utilisation de la structure topologique, de l'échéancier et de l'espace voxel.

Nous avons vu précédemment que la structure hiérarchique décrivant la topologie et la géométrie de la plante au cours de la simulation comporte un niveau *plante* (Cf. Fig.47). Nous allons utiliser ce niveau pour faire croître un ensemble de plantes et nous traiterons la compétition pour l'espace entre les individus à l'aide de la technique de l'espace voxel. Ainsi, la figure 63.a montre un bosquet d'Epiceas vu de dessus, un

arbre central étant entouré par six autres. On peut voir que chaque arbre se développe tout en évitant les contacts avec ses voisins. On remarquera que c'est l'arbre central qui a le plus de mal à se développer, gêné qu'il est par tous les autres. La figure 63.b nous permet de comparer l'architecture d'un arbre en croissance libre à celle de l'arbre gêné au milieu du bosquet.

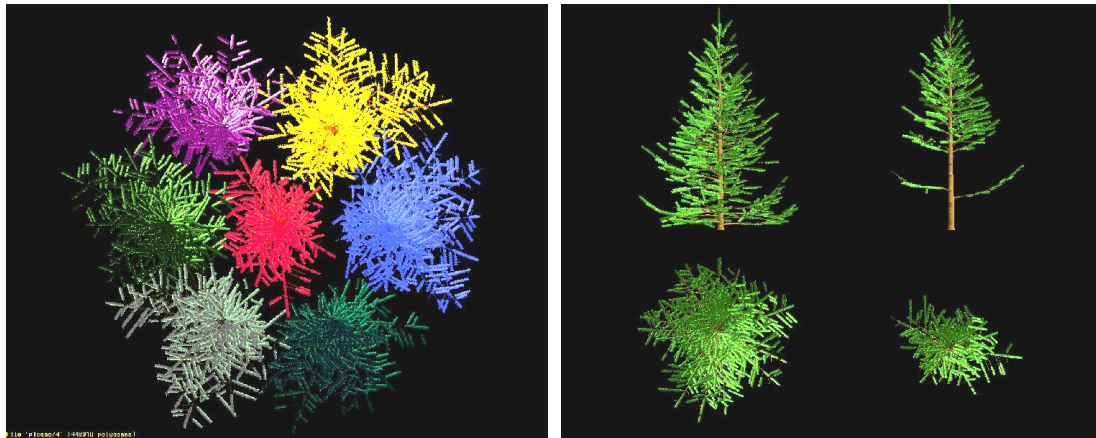


Figure 63 Simulation d'un bosquet d'épicéas avec gestion de la compétition pour l'espace (Blaise, 1991) : (a) vue de dessus du bosquet, (b) vue de dessus et de face d'un épicéa en croissance libre (à gauche) et de l'arbre central du bosquet (à droite).

Cette simulation peut être comparée à celle réalisée à l'aide de la méthode des L-systèmes utilisée par Prusinkiewicz (Cf. Fig.43-44). Les résultats obtenus sont similaires bien que, dans le cas des L-systèmes, ce soit la compétition pour la lumière qui génère ces couronnes d'arbre déformées, et non pas la compétition pour l'espace.

5 ARCHITECTURE DES PLANTES ET PAYSAGISME

5.1 DES OUTILS DE CAO POUR LA CONSTRUCTION DES PLANTES

Dans un chapitre précédant (Cf.4.1), nous avons présenté différents modèles informatiques utilisés pour la représentation des végétaux. Si ces approches sont de nature très différente, la plupart d'entre elles ont en commun le fait qu'elles introduisent une notion de dynamique dans le modèle. C'est d'ailleurs souvent la simulation de la croissance de la plante qui complique la tâche des chercheurs. C'est pourquoi certains travaux se sont intéressés plus spécifiquement à la construction des plantes en tant qu'objet 3D, se rapprochant ainsi de la CAO (Conception Assistée par Ordinateur).

5.1.1 Le logiciel TREE PROFESSIONAL (*Onyx Computing*)

5.1.1.1 Concepts généraux

Le logiciel TREE PROFESSIONAL de la société ONYX (Onyx computing, 1995) se présente comme un modèle paramétrable dédié à la production de plantes en 3D. Les modèles sont des logiciels couramment utilisés qui permettent de construire des objets quelconques en 3D à condition d'être quelque peu habile.

Le logiciel TREE se veut différent dans la mesure où il n'est pas nécessaire d'être biologiste ou botaniste pour construire une plante. Le logiciel fournit à l'utilisateur des plantes de référence (Fig.64) issues d'une librairie qu'il peut ensuite modifier à loisir en fonction du résultat recherché par modification de la valeur de paramètres essentiellement géométriques (longueur des axes, angle de branchement...) et graphiques (forme des feuilles et des entre-nœuds, couleurs et textures...) (Fig.65).

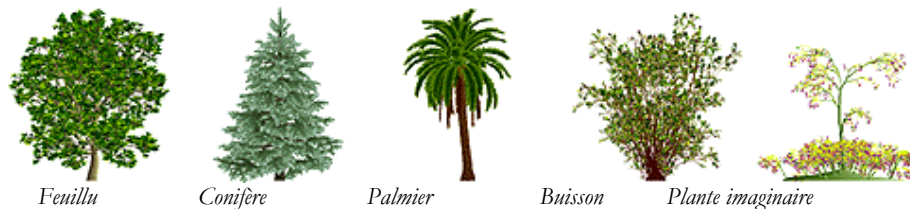


Figure 64 Arbres de références de la bibliothèque du logiciel TREE.

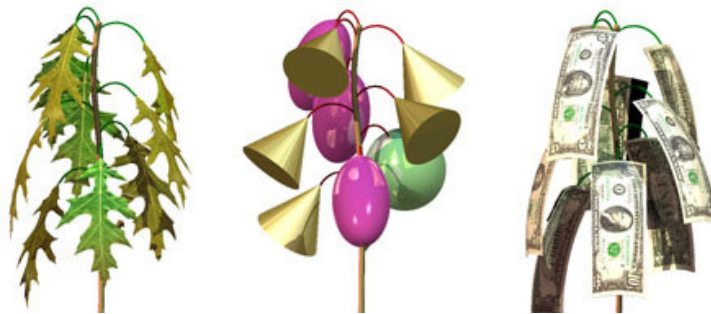


Figure 65 Plante à feuilles, à cloches et à dollars...

La topologie de ces plantes de référence est générée par le processus de récursion structurée (Bosanac, 1990). Le végétal est considéré comme une structure hiérarchique à 4 niveaux : le tronc, les branches maîtresses, les rameaux et les brindilles. Chaque entité, en fonction de son niveau dans la hiérarchie et de ses paramètres de fonctionnement, pourra donner naissance à des entités du niveau suivant. On notera que cette approche sous-entend que les plantes sont auto-similaires (Cf.4.1.1) et qu'elle pourrait être formalisée sous forme de L-système (Cf.4.1.2). La géométrie, quant à elle, est semblable à celle définie par le modèle de Fisher et Honda (Cf.4.1.1) mais est complétée par des notions plus complexes (courbure des branches, excentricité de la couronne...).

5.1.1.2 Des outils spécialisés

Le modèleur de végétal TREE PROFESSIONAL intègre également des outils spécifiques afin, par exemple, de modifier interactivement la forme des troncs (Fig.66). Il est possible également de simuler une taille mécanique de branches ou du tronc (Fig.67).



Figure 66 Déformation de la forme des troncs.



Figure 67 Simulation de la taille.

Un module spécialisé, externe au logiciel TREE PROFESSIONAL mais compatible, TREE EIAS STORM a été développé afin de simuler l'effet mécanique du vent sur la forme des plantes (Fig.68). Une interface graphique permet de définir les caractéristiques mécaniques de chaque entité dans la structure hiérarchique (tronc, branches...) et de paramétrer le vent (direction, force).



Figure 68 Cocotiers dans la tempête.



Figure 69 Pines de Angela Perkins.

Un autre module, TREE PAINTER, permet de composer des scènes avec des plantes en 2D. La figure 69

en est un exemple : cette image est une réplique d'un paysage réalisé à l'aide de la chaîne de logiciels AMAP (Cf.5.2).

5.1.1.3 Quelques réalisations

TREE PROFESSIONAL et des modules comme TREE EIAS STORM sont vendus dans le monde entier sur plate-forme Mac et PC. Leurs utilisateurs génèrent ainsi des végétaux de belle facture afin de les intégrer dans des scènes plus complexes pour des domaines applications variés : art (Fig.70, 71, 72, 73), jeux vidéo (Fig.74, 75), plan d'aménagement (Fig.76, 77). Par contre, le résultat final (l'image de synthèse) est obtenu par le biais d'autres logiciels spécialisés dans le rendu, capables de lire des fichiers de description géométrique issus de TREE (via le format DXF par exemple).



Figure 70 *Dino* de Bau Struye (USA).



Figure 71 *Amazonia Evidence* de David Argemi (Espagne).



Figure 72 *Cave* de Uwe Walter (Chine).



Figure 73 *Oath* de Uwe Walter (Chine).



Figure 74 Image extraite du jeu *Gilbert and the Chemistycal Island* de Hans Olsson (Sweden).

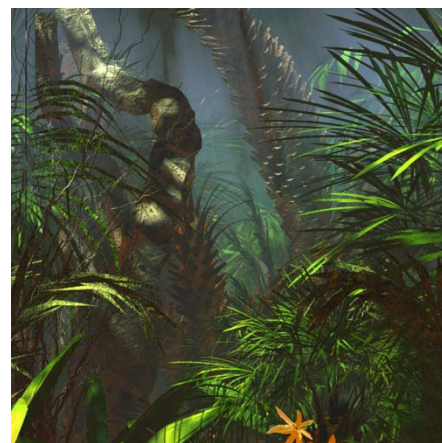


Figure 75 Image extraite du jeu *Congo* de Ethan Summers (USA).



Figure 76 Hiyoshihoncho Entrance de Yoshiteru Enomoto (Japon).



Figure 77 Tanmachikouen de Yoshiteru Enomoto (Japon).

5.1.2 Le logiciel XFROG (*Greenworks*)

5.1.2.1 Concepts généraux

Récemment, la jeune compagnie allemande GREENWORKS (Greenworks, 1996) a développé un modèleur procédural interactif pour générer des entités biologiques : le logiciel *xfrog*. Ce dernier est présenté comme une nouvelle technique interactive pour la modélisation orientée objet, basée sur une structure de graphe.

Ainsi, le buisson, sur la gauche de la figure 78, a été généré à partir du graphe visualisé sur la partie droite de cette figure. L'icone *appareil photographique* représente la racine du graphe ; il crée toutes les structures de données nécessaires pour les autres icônes. Le second nœud du graphe (le symbole circulaire) indique que ses successeurs dans le graphe seront multipliés. En l'occurrence, des branches seront créées suivant la topologie indiquée par le symbole arbre stylisé : une branche portant 2 autres branches. La ligne double connectant deux nœuds du graphe indique une récursion et le symbole feuille définit une géométrie simple pour les limbes.

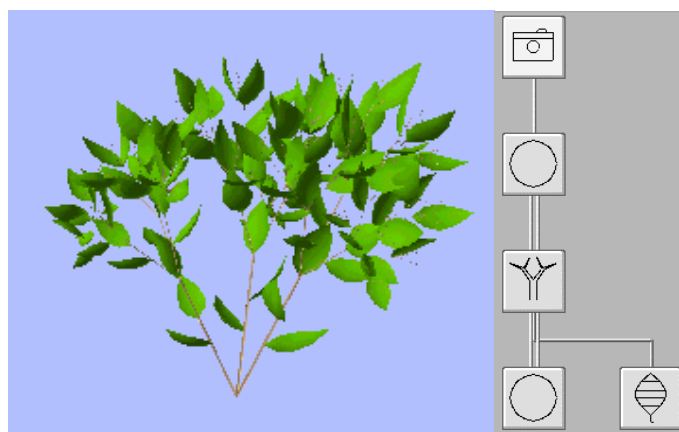


Figure 78 Le logiciel *xfrog* : génération d'un buisson.

La technique de modélisation employée peut être considérée comme un éditeur graphique (Lintermann et Deussen, 1998) dans lequel on définit les règles de génération des objets, règles comparables aux règles de production des L-systèmes (Cf.4.1.2).

Chaque icône disponible (dix en tout) pour constituer le graphe est identifiée par des caractères spécifiques : géométrie, propriétés visuelles du matériau... D'autres composants permettent de reproduire ces primitives de base par algorithme afin de pouvoir aisément générer des formes complexes telles les

spirales phyllotaxiques. Il existe également des composants qui permettent des opérations complexes sur des structures ou la génération de formes spéciales. Les divers composants peuvent être classifiés par fonctionnalité :

- génération de formes géométriques complexes : surfaces de révolution, grands volumes, objets naturels (feuille, fleur),
- duplication et positionnement particulier de composants : spirales phyllotaxiques,
- définition de paramètres globaux : sources de lumière, champs de gravité, photo et gravitropisme,
- transformations géométriques appliquées à des sous-structures du graphe.

5.1.2.2 Quelques réalisations

Comme le logiciel TREE PROFESSIONAL (Cf.5.1.1), *xfrog* génère des formes naturelles mais n'assure pas le rendu final. En fait, en sortie, il crée des fichiers à des formats spécifiques (DXF, Inventor/VRML et Alias/wavefront) utilisables par des logiciels spécialisés dans le rendu (RayShade et POV-Ray par exemple). Nous verrons dans les figures suivantes (Fig.79 à 88) que le résultat est remarquable et que *xfrog* ne se restreint pas au domaine des végétaux.

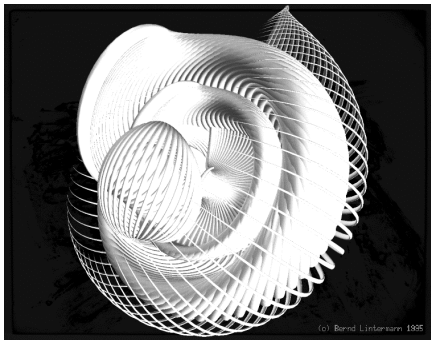


Figure 79 Forme naturelle.



Figure 80 Une mouche...



Figure 81 Cactus en fleur.



Figure 82 Rose.



Figure 83 Pommes et ananas.

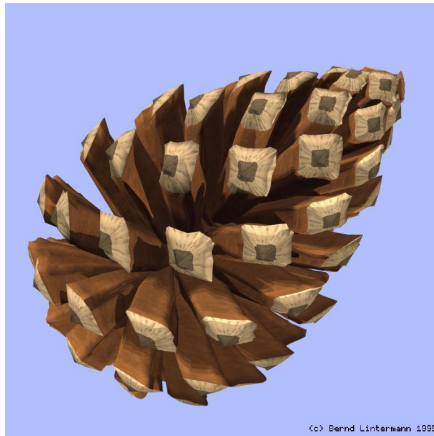


Figure 84 Pomme de pin.



Figure 85 *Pinus*.



Figure 86 Épicéa.

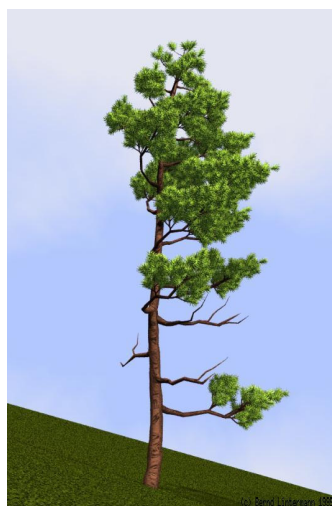


Figure 87 *Pinus Mugo*.



Figure 88 Peuplier.

5.2 UNE APPROCHE BOTANIQUE POUR LE PAYSAGISME : AMAP INTEGRAL

5.2.1 Des outils de modélisation pour le paysage

La représentation des plantes dans l'image de synthèse se limite le plus souvent à une forme simplifiée, sans grand réalisme botanique, tout simplement parce qu'elles ne constituent pas le centre d'intérêt principal. Par contre, si l'on considère la tendance actuelle des projets d'aménagement à faire de plus en plus appel aux technologies informatiques afin de mieux anticiper l'évolution du projet et des paysages, les plantes ne doivent plus être considérées sous une forme figurative mais bien sous une forme réaliste, conforme aux lois de la botanique. Si ce type de demande fut longtemps l'apanage de projets de prestige à cause du coût de production, la démocratisation de l'informatique, autant sur le plan matériel que logiciel, et l'extraordinaire augmentation des performances des micro-ordinateurs font que l'image de synthèse est maintenant un outil qui tend à remplacer les dessins et les plans classiques dans l'illustration des projets d'architecture et de paysagisme.

Ainsi, AMAP a développé une gamme de logiciels, regroupés sous le nom d'Ateliers de Modélisation pour l'Aménagement du Paysage (AMAP Intégral), dédiée à la simulation dans l'espace et le temps de projets d'aménagement et commercialisée par la société Bionatics (Bionatics, 2000) :

- AMAP Genesis : un moteur de croissance permettant de simuler la croissance de n'importe quelle espèce végétale en 3D,
- AMAP Altis : un modelleur numérique de terrain capable de créer la topographie d'un relief ou de le recréer à partir de données mesurées,
- AMAP Orchestra : un logiciel de mise en scène et de rendu de grands passages.

Ces modules s'accompagnent d'outils spécifiques de création d'objets graphiques (feuille, fruit, pot, vase, chaise ...), de définitions des caractéristiques des matériaux (couleur et aspect des objets) (Cf.3.3), de créations et de plaquage de textures (Cf.3.4), d'importation et d'exportation d'objets en divers formats (DXF, OBJ...).

5.2.2 AMAP Genesis : un modelleur botanique

5.2.2.1 Principe de fonctionnement

Le moteur de croissance Genesis est l'héritier d'un prototype qui avait été mis au point en Côte d'Ivoire dans les années 70. Ce prototype, développé par Philippe de Reffye, utilisait un ordinateur Hewlett-Packard. Bien que spécifiquement dédié à l'étude des modèles architecturaux définis par Hallé et Oldeman et plus particulièrement au caféier, il est à l'origine de tous les moteurs de croissance qui seront développés plus tard au sein de l'AMAP. Toutefois, ce logiciel de simulation possédait de nombreux défauts de jeunesse :

- Le niveau de détail botanique pris en compte était insuffisant.
- Le programme informatique était mal structuré et écrit dans un langage de programmation obsolète (le HPL), valable uniquement pour les ordinateurs Hewlett-Packard. A cette époque les micro-ordinateurs n'avaient pas fait leur apparition.
- Les données décrivant le modèle architectural à simuler étaient définies au sein même du programme rendant ainsi peu interactive l'utilisation du moteur.
- L'affichage graphique des plantes générées n'était possible que sur table traçante.

Des améliorations conceptuelles du modèle de base et la réécriture d'une nouvelle version du programme se sont imposées et ont permis la création du premier logiciel AMAP aux normes de l'informatique moderne (Jaeger 1987 ; de Reffye et *al.* 1988).

En particulier, toutes les différentes générations de moteur qui se sont succédées ont conservé le même principe de fonctionnement général :

- En entrée du programme, on trouve un fichier paramètre décrivant la morphologie et le fonctionnement de la plante à simuler. Ce fichier peut être créé et édité via un logiciel de saisie adapté à la génération du moteur utilisé.
- Le résultat de la simulation est un fichier, appelé *ligne élastique*, contenant la description géométrique de la plante générée. Chaque entité constituant le végétal (entre-nœud, feuille, fleur ou fruit) y est défini en indiquant sa nature et sa géométrie (dimensions et position dans l'espace). Elles ne sont donc pas décrites sous forme de polygones ou facettes, ce qui allège considérablement la taille des fichiers générés.
- La visualisation des lignes élastiques se fait en utilisant des objets graphiques 3D, préalablement définis dans une bibliothèque de symboles et associés aux différents éléments de la ligne élastique par l'intermédiaire d'un fichier de description.

5.2.2.2 *Structuration des données botaniques*

Les réflexions des agronomes et informaticiens du CIRAD menées conjointement avec les chercheurs de l'Institut de Botanique de Montpellier sur l'architecture des plantes ont permis de définir plus précisément la pertinence et la hiérarchisation des paramètres du modèle, et de mieux les structurer. Ce travail a débouché sur une organisation précise des fichiers paramètres décrivant l'architecture des plantes, toujours utilisée par le logiciel Genesis.

Le concept de base contenu dans un fichier paramètre est la notion de structure végétale : c'est une architecture végétale simple très proche des modèles architecturaux. Parmi les types de paramètres ou variables définissant une telle structure, on distinguera :

- Les variables globales : celles-ci définissent des caractéristiques générales de la structure végétale comme le type de modèle architectural, les modalités de l'élagage des branches mortes... On définit aussi des paramètres qui modifient globalement la géométrie de la plante : le redressement orthotrope en bout de branche, la variation de l'angle de branchement des branches le long du tronc ou les lois de croissance en épaisseur des branches. Les paramètres qui contrôlent la stratégie de réitération se situent également à ce niveau.
- Les variables qui concernent les organes : celles-ci permettent de positionner la présence des différents organes (feuilles, fruits) en fonction de leur date d'apparition et de leur durée de vie.
- Les variables par ordre : chaque ordre de ramification peut avoir ses propres caractéristiques qu'il faut renseigner. On distingue :
 - les variables de type topologique : par exemple le nombre d'axillaires par nœud, le positionnement de la première floraison sur un axe, le nombre d'entre-nœuds par unité de croissance, les notions d'axe long et d'axe court pour simuler sur un même ordre des bourgeons d'âge physiologique différent.
 - les variables de type fonctionnel : ce sont les lois de probabilités qui concernent la croissance, la mortalité et la ramification, les rapports de rythmes.
 - les variables de type géométrique : les longueurs d'entre-nœuds, les angles de branchement et de phyllotaxie, les modules d'élasticité appartiennent à cette classe.

Une structure simple permet d'aborder de nombreuses espèces végétales (Fig.89). Les possibilités restent toutefois limitées. Si on peut simuler la plupart des modèles architecturaux, on reste tout de même contingenté par le modèle qui ne peut pas réitérer. Ces limitations ont donc conduit à des améliorations dans la prise en compte des données botaniques.

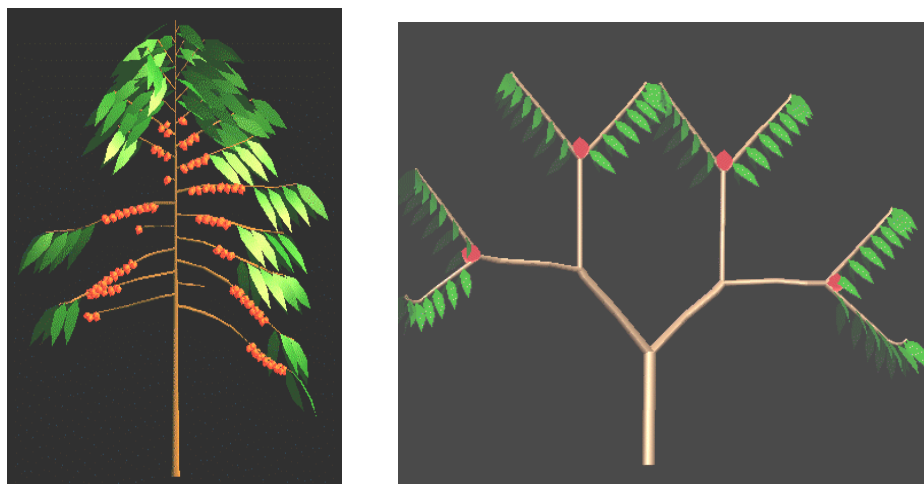


Figure 89 Modèles architecturaux et structures simples : Roux et Leeuwenberg.

- ◆ Si le prototype du moteur de croissance n'autorisait la définition que d'une seule structure végétale par plante, Genesis permet d'empiler plusieurs types différents de structures sur un même végétal : la

réitération, les structures surnuméraires, les organes procéduraux (feuilles composées et fleurs).

- La réitération représente la possibilité pour une plante de dupliquer son modèle architectural de base afin d'augmenter son potentiel de développement. En informatique, on qualifierait la réitération de processus récursif. Les plantes obtenues en activant ce phénomène de réitération sont en fait des colonies de la structure simple de base qui s'exprime naturellement au stade jeune (Fig.90).

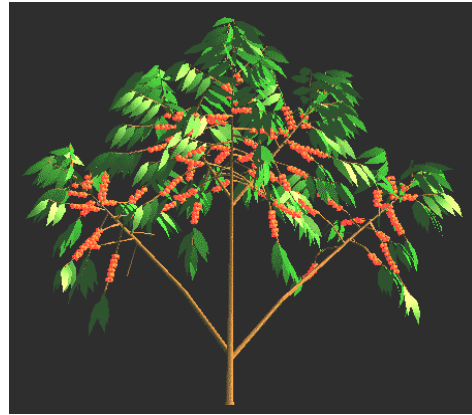


Figure 90 Réitérations d'une structure simple.

- Les structures surnuméraires permettent de construire des architectures végétales composites, constituées d'un empilement de modèles architecturaux différents, ou du même modèle architectural comme dans le cas de la

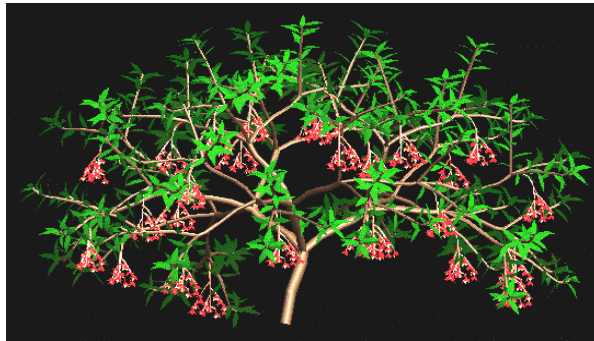


Figure 91 Structure végétale et inflorescences : une architecture composite.

réitération. Une structure végétale portant des inflorescences est une architecture composite (Fig.91). En effet, une inflorescence est en elle-même une véritable petite architecture végétale avec son propre paramétrage. Par ailleurs, la vitesse de développement de chacune de ces structures est paramétrable et définie par rapport à celle de la structure principale.

Associé à la réitération, les structures surnuméraires permettent de générer des formes architecturales très complexes et très variées. En particulier, la gestion dans le temps de la succession des différentes structures permet de simuler des phases caractéristiques de la croissance du végétal comme l'acrotonie ou le vieillissement. En fait, cette utilisation de la notion de structure surnuméraire préfigure les notions d'âge physiologique et d'axe de référence (Fig.92).

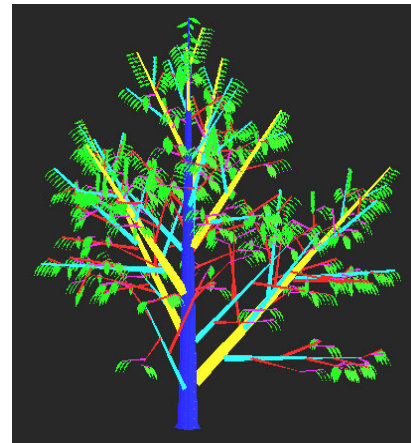


Figure 92 Empilement de structures et âge physiologique.

- De même que les empilements de structure permettent de générer des architectures végétales complexes, les organes procéduraux autorisent la description d'organes dynamiques, non statiques dans le temps, suivant leurs propres lois de développement. On utilise de tels organes lorsque l'on souhaite détailler une morphologie complexe et simuler son évolution dans le temps. C'est souvent le cas pour les palmes ou les inflorescences (Fig.93).

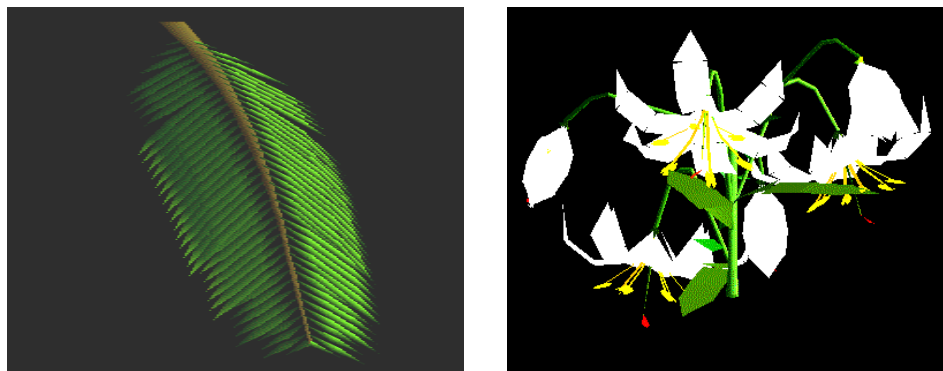


Figure 93 Organes procéduraux : (a) palme, (b) inflorescence.

◆ Sur le plan de la morphologie, dans le prototype, l'unité de croissance était définie très sommairement comme un module composé d'un nombre de métamères fixes, se répliquant d'une façon rythmique le long de l'axe. Seul le dernier métamère pouvait porter une branche. Dans la version actuelle, pour chaque ordre chaque entre-nœud appartenant à une unité de croissance est référencé et peut porter différents types de bourgeons axillaires qui engendrent soit des rameaux longs ou courts pour simuler l'acrotonie (Fig.94), soit des structures secondaires ou des organes. On simule le vieillissement d'un axe en diminuant progressivement le nombre d'entre-nœuds formés par unité de croissance (phénomène de dérive). On simule encore la limitation du développement des premiers étages de branches mis en place sur le tronc (effet de base). Enfin, divers paramètres permettent de positionner les organes selon leurs types et leurs durées de vie.



Figure 94 Simulation d'une unité de croissance avec de l'acrotonie : (a) été, (b) hiver.

◆ Sur le plan de la géométrie, on peut contrôler les longueurs de chaque entre-nœud, les modifications des angles de branchement et de phyllotaxie, l'affaissement des branches et le géotropisme. Une typologie des types d'insertion des branches en fonction de l'ordre permet de simuler diverses orientations couramment rencontrées pour les axes et les organes.

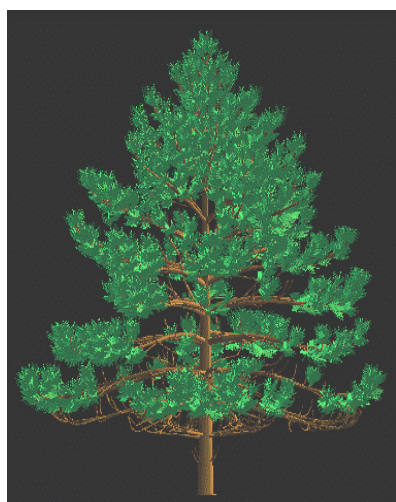


Figure 95 Une structure simple : le pin.

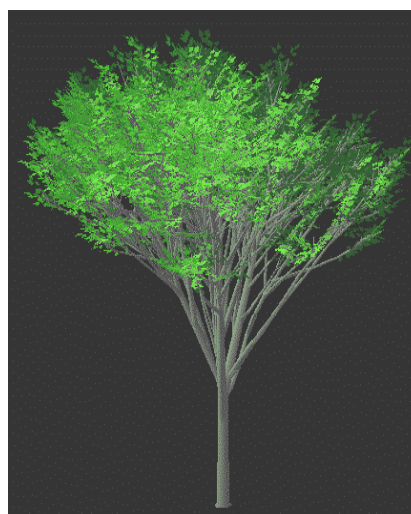


Figure 96 Des structures empilées : le kéaki.

Ces améliorations permettent de simuler des arbres comme le pin (Fig.95) au moyen d'une seule structure. Toutefois, les architectures sympodiales des grands arbres comme l'orme du Japon (Fig.96) nécessitent, pour être simulées de façon réaliste, l'empilement de plusieurs structures.

5.2.2.3 *Création d'une plante virtuelle*

La conception d'une nouvelle plante virtuelle nécessite de nombreuses observations du végétal en milieu naturel avant de pouvoir le visualiser sur un écran graphique. Ces observations aboutissent à la création d'un fichier paramètre qui permet au moteur de croissance de générer un fichier, la ligne élastique, contenant la description géométrique du végétal mesuré. Enfin, on peut le représenter en utilisant des symboles graphiques prédéfinis, associés aux différentes entités de la plante selon leur nature.

◆ *Observations et mesures*

Des photographies in situ, des croquis et différentes mesures par sondage sur l'architecture permettent d'obtenir des informations pertinentes sur la morphologie et le mode de croissance de la plante. La collecte des données s'effectue en partant de considérations générales avant de s'intéresser à des caractères botaniques plus fins.

Dans un premier temps, on définit le type de modèle architectural, les modalités de la réitération et la stratégie d'élitage. Il faut à ce stade décider si une ou plusieurs structures sont nécessaires pour décrire la plante. Ensuite, on détermine les différents types d'axes en fonction de leur orientation résultant des angles de branchement, de la phyllotaxie, de la flexion mécanique et des différents tropismes (géotropisme, phototropisme). Enfin, on étudie plus particulièrement la morphologie des différents types d'unités de croissance, selon les ordres de ramification, en notant leur acrotonie et la forme, les dimensions et le type d'orientation des organes (entrenœuds, feuilles et fruits).

Rappelons que le but recherché consiste à produire des modèles de plantes suffisamment réalistes afin de pouvoir les utiliser dans des applications paysagères. Si on reste dans un contexte de qualité botanique, on n'en oublie pas pour autant le rendu graphique qui doit être performant. Un critère important de la qualité d'un rendu en image de synthèse réside dans la vitesse d'affichage des objets générés. Autrement dit, ces objets se doivent de limiter le nombre de polygones ou facettes nécessaires à leur affichage, mais tout en gardant un modèle de qualité. C'est pourquoi, les mesures effectuées sur la plante doivent privilégier les caractéristiques majeures permettant de l'identifier en tant qu'espèce, plutôt que des détails dont la prise en compte ne se solderait que par un alourdissement du fichier généré. Ce type d'expertise, indispensable mais rare, nécessite de bonnes connaissances en botanique et un savoir-faire certain dans la mise au point du fichier paramètre.

◆ *La saisie des données*

Une fois toutes les données qualitatives et quantitatives rassemblées, il faut construire le fichier paramètre qui sert au moteur de croissance. La création d'un tel fichier s'effectue à l'aide d'un programme spécifique d'édition qui permet de renseigner tous les paramètres nécessaires à la simulation. Bien que développé il y a une dizaine d'années, cet éditeur n'a guère évolué dans son interface et reste réservé à l'usage des seuls initiés. Ce manque de convivialité résulte d'un choix délibéré. En effet, la philosophie d'AMAP consiste à dire que seuls des spécialistes sont capables de renseigner un fichier paramètre en vue de générer une plante botaniquement correcte. Il est vrai que le nombre et la spécificité des paramètres définissant un végétal sont tels qu'il est difficilement envisageable de demander à un utilisateur des logiciels AMAP de s'investir dans une formation qui serait trop longue et trop spécialisée. Seule une perte de productivité en résulterait. Toutefois, nous verrons dans le paragraphe suivant que le logiciel Genesis permet d'accéder à certains paramètres décrivant la plante et de les modifier à l'aide d'un « mini-éditeur ».

5.2.2.4 *Fonctionnalités du logiciel Genesis*

AMAP Genesis est un moteur de croissance procédural de végétaux. Il est capable de simuler à n'importe quel âge un végétal en 3 dimensions en tenant compte de la morphologie naturelle de la plante (Fig.97). Un grand nombre de fonctionnalités permettent de contrôler l'esthétique et la simplification du végétal,

son comportement botanique au cours du temps, de le tailler, de paramétrer les saisons, les couleurs et les textures des feuilles et des troncs (Fig.98), de contrôler le nombre de polygones exportés en fonction de la distance et de générer des géométries différentes pour un même végétal.

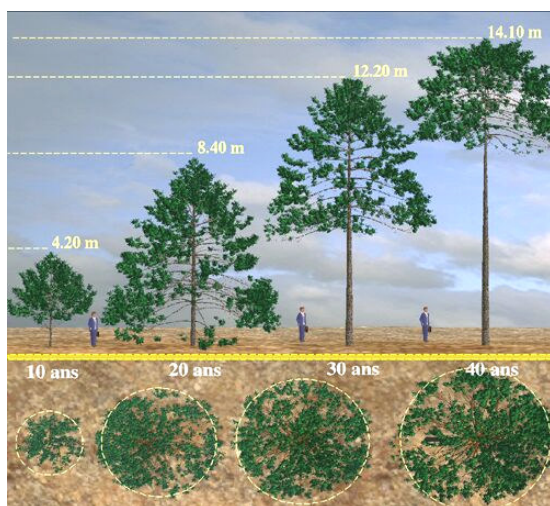


Figure 98 Simulation de croissance du pin noir.



Figure 97 Inflorescence de marronnier avec textures.

Nous allons maintenant brièvement énoncer ses principales fonctionnalités.

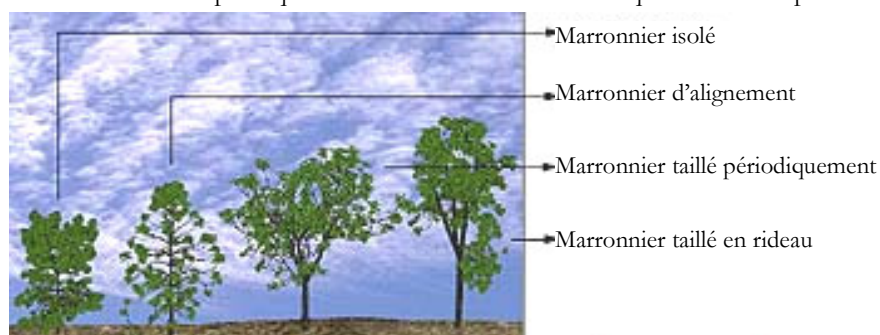
o *Une simulation de croissance botaniquement cohérente*



Comme nous l'avons déjà précédemment évoqué, la méthodologie AMAP est basée sur la botanique. Aussi, les plantes calculées en 3 dimensions par le logiciel AMAP Genesis sont botaniquement cohérentes dans leur architecture et leur modèle de croissance.

o *Simulation de conditions particulières*

AMAP est capable de simuler la croissance d'une plante dont le développement est affecté par des conditions particulières. Ainsi, pour certaines espèces, plusieurs fichiers paramètres sont disponibles afin de pouvoir simuler des formes de couronne spécifiques en fonction des traitements qu'aura suivi la plante. Le logiciel peut alors être renseigné différemment sur les paramètres morphologiques de croissance et de développement d'une plante. Des simulations de tailles périodiques peuvent également être reproduites à partir des



mesures effectuées sur le terrain.

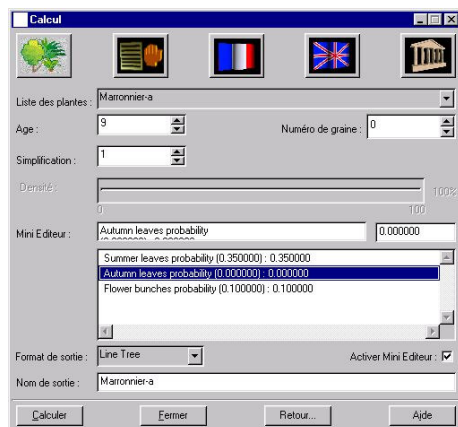
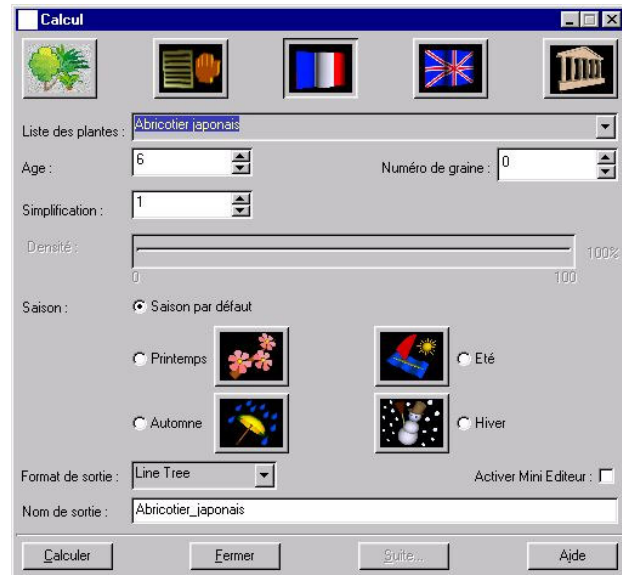
o Barre des menus

Afin de faciliter la convivialité de l'interface utilisateur, les fonctions principales d'AMAP Genesis sont accessibles à partir d'une barre d'icônes : modes de visualisation, fenêtre des paramètres de calcul, contrôle des couleurs et options d'élagage ...



o Paramétrage du moteur de croissance

Après avoir sélectionné une plante par son nom en français, en anglais ou en latin, AMAP Genesis la simule automatiquement à l'âge choisi. Le numéro de graine aléatoire permet d'exploiter les possibilités de la simulation stochastique : à partir d'un même fichier paramètre, il est possible de générer des formes architecturales différentes pour une même espèce (apparition décalée des ramifications, modification de la géométrie des branches ...). Le niveau de simplification permet de générer des plantes dont le *poids* en terme de polygone est allégé en fusionnant des entre-nœuds qui se suivent le long des branches et qui ont une direction principale proche. Cette simplification se fera au détriment de la précision dans la courbure des branches.



L'utilisateur a également la possibilité de gérer lui-même certains paramètres décrivant la plante par l'intermédiaire d'un *mini-éditeur*. Ces paramètres très simples permettent toutefois de mieux adapter l'allure générale de la plante simulée à sa problématique. Les paramètres éditables sont variables et dépendent des choix qui ont été faits au départ par le concepteur du fichier paramètre. Ceci permet de garantir la qualité botanique des plantes générées par AMAP Genesis.



Hiver



Printemps



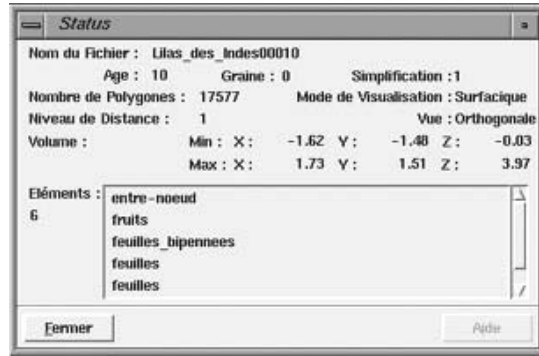
Eté



Automne

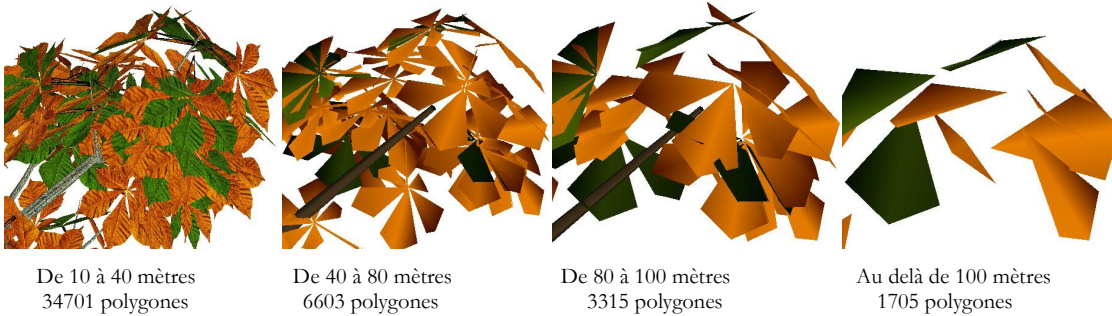
o *Boîte de status*

Une boîte de dialogue renseigne sur les dimensions exactes de ce Lilas des Indes à 10 ans, sur son nombre de polygones en niveau de détail 4 et sur le nombre de niveaux géométriques qui constituent le modèle 3D.



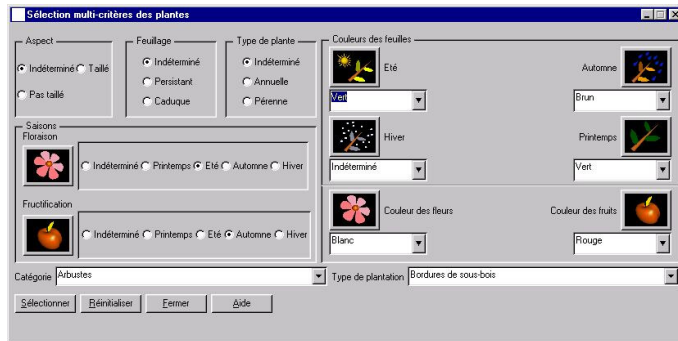
o *Simplification automatique par niveau de distance*

Nous avons précédemment vu qu'il était possible de simplifier le nombre de polygones utilisé pour la représentation de la plante en paramétrant le *niveau de simplification*. Il existe un second moyen, automatique celui-là, se basant sur la distance séparant la plante du point d'observation. Cette fonction permet de simplifier automatiquement une plante trop lourde en polygones sans détériorer son aspect esthétique à la distance correspondante. Voici un exemple concret effectué sur les branches basses d'un marronnier.



o *Système de requêtes multicritères botaniques*

La base de données des plantes d'AMAP comprend plus de 300 plantes. Une boîte de dialogue permet de sélectionner une plante en fonction de ses caractéristiques botaniques : période de floraison, couleur des feuilles, des fruits et des fleurs, type de plantation, ...



5.2.2.5 *Le catalogue des plantes virtuelles*

Bien que la création d'un fichier paramètre demande l'intervention d'un spécialiste, il semble possible de pouvoir générer tous les végétaux quelles que soient leur nature et leurs spécificités. Plusieurs centaines de plantes de tous les continents allant des plantes herbacées aux grands arbres en passant par les buissons et arbustes ont ainsi programmés pour les besoins de l'image de synthèse et des projets d'aménagement (Fig.99).

La conception d'un catalogue de plantes virtuelles aux normes de la botanique et de l'image de synthèse et répondant à la demande du marché n'est pas simple. Il faut modéliser des plantes aussi diverses que des herbes, des fleurs, des arbustes ou des arbres tempérés et tropicaux. En effet, des architectes japonais,

canadiens ou tahitiens n'ont pas les mêmes besoins. Il faut aller chercher les données dans les pépinières, les jardins et les parcs des pays concernés au cours de missions spécifiques. C'est à ce coût que se constitue peu à peu la bibliothèque de plantes AMAP qui est utilisée maintenant par plus de mille cabinets d'architectes et de sociétés intéressés par l'image de synthèse.

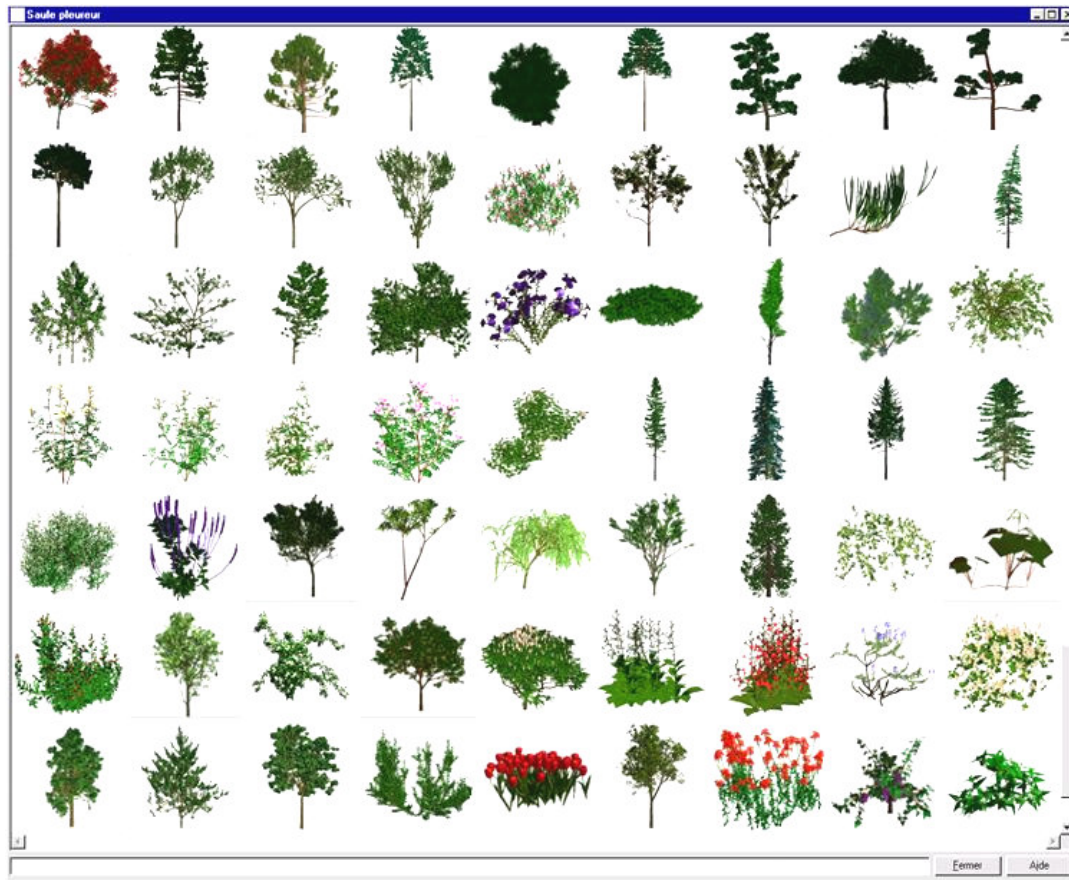


Figure 99 Extrait du catalogue de plantes virtuelles AMAP.

La base de données végétale virtuelle constituée vise la représentation réaliste des plantes et non leur production. Elle concerne des sujets aussi divers que la promotion d'un genre comme les bégonias ou la représentation des plantes fossiles.

o *Buissons, baies*



Corète du Japon

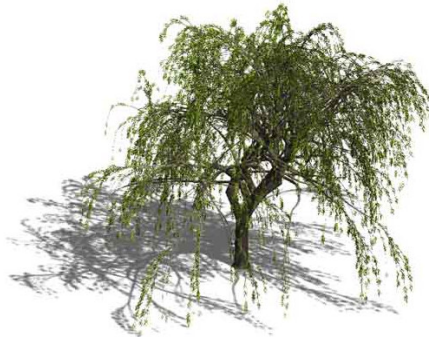
Forsythia ou mimosa de Paris

Weigéla

o *Arbres*



Marronnier



Saule pleureur

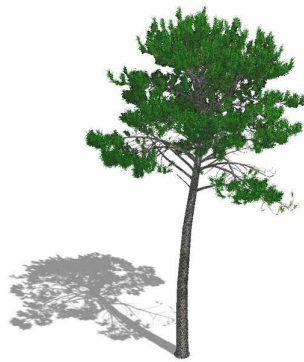


Tilleul de Hollande

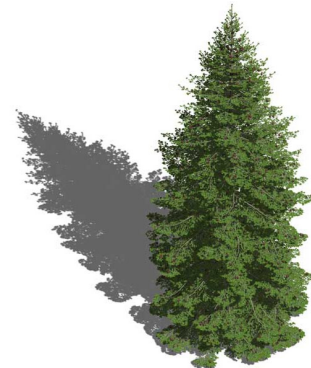
o *Pins, conifères*



Mélèze d'Europe



Pin maritime



Sapin de Vancouver

o *Palmiers*



Cocotier

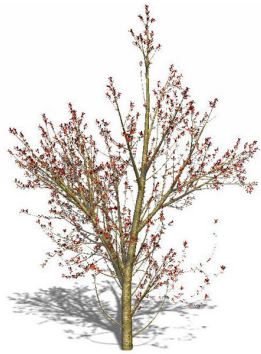


Palmier à tronc rouge

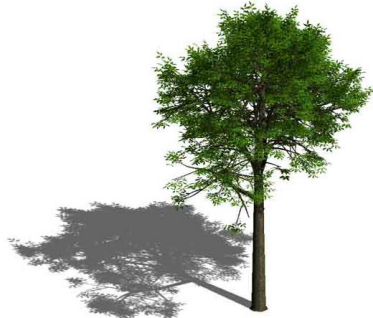


Palmier chamaerops

o *Arbres fruitiers, fruits*



Cerisier japonais à fleurs

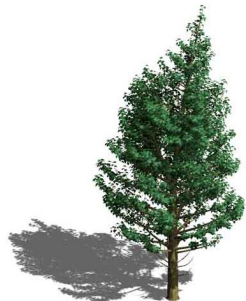


Châtaignier



Olivier de Bohême

o *Arbustes*



Houx



Arbre à perruques



Bois-joli ou bois-gentil

o *Plantes grimpantes*



Clématite



Hortensia grimpant

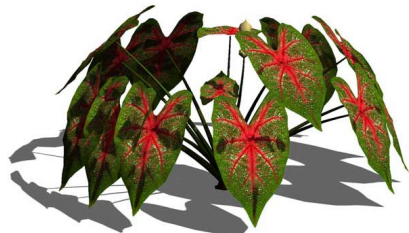


Lierre rampant

o *Plantes d'intérieur*



Ficus benjamina



Palette de peintre



Palmier d'Arec

o Fleurs



Tulipe hybride de Darwin



Sauge bleue

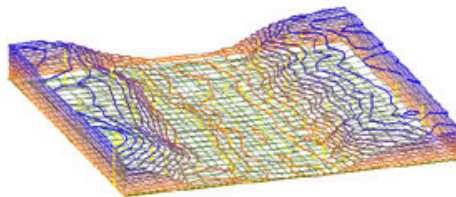


Orchidée (schilleriana)

5.2.3 Conception et visualisation d'un paysage virtuel

5.2.3.1 Création d'un terrain : AMAP Altis

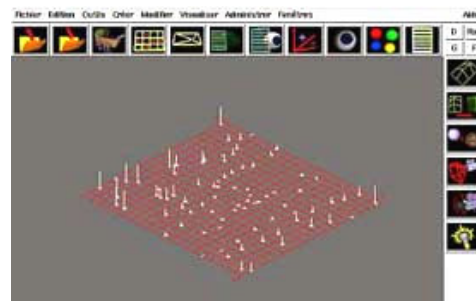
Altis est un modelleur numérique de terrains, capable de créer la topographie d'un relief ou de la reconstituer à partir d'un fichier de points de géomètre. Il a été développé dans le souci de répondre, dans le domaine de l'imagerie, aux contraintes de modélisation rapide et précise de terrains de toutes tailles. Il



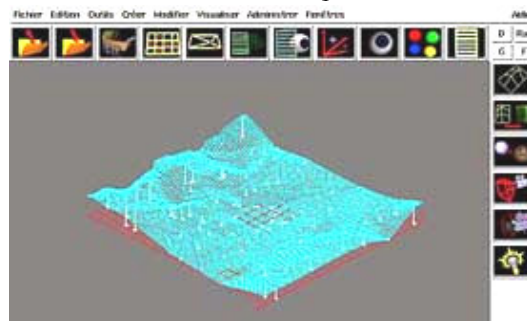
est capable de trianguler, d'interpoler une grille, de déformer un maillage par un autre, de calculer des courbes de niveaux et de délimiter des parcelles pour l'affectation de zones à partir de plans ou de cartes. Sa capacité à projeter et intégrer un contour au maillage d'un terrain permet d'affiner la délimitation de parcelles pour l'affectation de groupes d'objets, de couleurs et de textures. Un rendu surfacique temps réel donne une première visualisation du terrain avec les ombrages.

o Récupération de fichiers de points de géomètres

Le module topographique du logiciel AMAP INTEGRAL permet de récupérer des fichiers de points de géomètres x, y, z et de reconstituer un terrain en 3D de manière simple et conviviale. Un mode création permet également de rentrer à l'aide de la souris des coordonnées de points et constituer ainsi un terrain directement dans AMAP. L'affichage d'un plan topographique en fond permet également de reproduire des courbes de niveau en 3D.



o Création d'un modèle numérique de terrain



Un maillage du terrain permet de constituer un modèle numérique de terrain en 3D. AMAP donne la possibilité de calculer cette maille suivant des définitions et des modes d'interpolation différents.

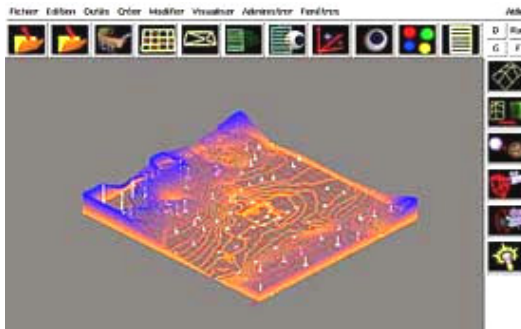
Des opérations de modification sont ensuite possibles après triangulation :

- La modification automatique applique des déformations d'un terrain à l'aide d'un autre relief.
- La modification manuelle permet de déformer le

terrain manuellement.

- La modification par zone permet la création de parcelles.

o *Création de courbes de niveaux*



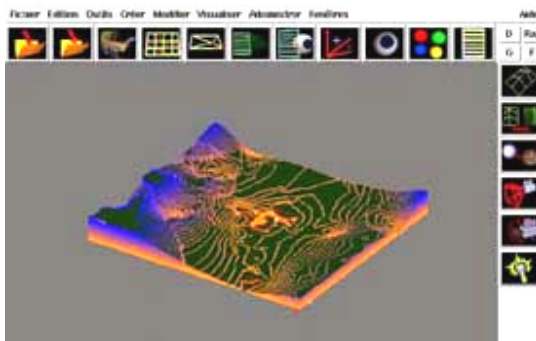
A partir du modèle numérique de terrain 3D, AMAP peut calculer les courbes de niveaux qui s'y rapportent. On peut définir le pas et l'altitude de départ du calcul.

Un dégradé de couleur permet de rendre lisible les altitudes.

o *Affichage d'un fond de plan ou d'une carte*

Cette boîte de dialogue permet à l'opérateur de positionner une image ou une carte en fond de plan et travailler par-dessus.

On peut alors reproduire un modèle numérique de terrain en 3D à partir d'un fond de plan scanné (parcelles, courbes de niveaux, plan topographique, cadastre...)



o *Affichage d'un terrain en mode surfacique*

Un rendu surfacique permet de donner une première visualisation du relief en distinguant le relief. On peut superposer un relief à ses courbes de niveaux ou visualiser plusieurs terrains en même temps ayant été mis à des altitudes différentes.

5.2.3.2 Conception d'un paysage : AMAP Orchestra

AMAP Orchestra peut être considéré comme le *chef d'orchestre* de la chaîne de logiciels AMAP : il permet de réunir terrains, architectures, infrastructures et végétations dans une même virtuelle. Une de ses particularités par rapport à d'autres logiciels existants est qu'il n'est pas limité par le nombre de polygones composant la scène à traiter. Il est en effet capable d'intégrer, de mettre en scène et de visualiser des maquettes virtuelles de grands paysages.

AMAP Orchestra récupère dans un format qui lui est propre les éléments issus des modules Genesis et Altis, précédemment présentés. Mais il peut également intégrer des éléments (architectures 3D, contours et groupes d'objets) venant d'AutoCAD V14 grâce à un plug-in spécialement développé. Ce dernier rend également possible l'intégration de courbes de niveaux provenant d'AutoCAD dans le module Altis.

D'autres fonctionnalités avancées lui permettent de positionner rapidement une grande quantité d'information 3D. Des effets spéciaux pour le rendu peuvent être créés très simplement, incluant des ombres portées, des réflexions surfaciques et des effets de brume.

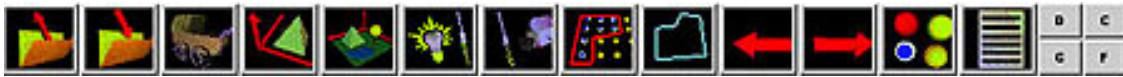
o Une capacité de calcul adaptée aux besoins des projets d'aménagement

Construire et manipuler des projets d'aménagement virtuels nécessitent des ordinateurs puissants et des logiciels performants. Une plante est représentée en trois dimensions par une quantité importante de polygones. Un marronnier modélisé à 25 ans peut ainsi représenter jusqu'à 700 000 triangles. L'affichage à l'écran de cette information gigantesque pose de réels problèmes techniques lorsque les maquettes contiennent plusieurs milliers d'objets 3D. AMAP Orchestra a été spécialement conçu pour pouvoir assurer le calcul et l'afficher de grandes quantités d'informations 3D. En particulier, il utilise des techniques informatiques telles la méthode du Z-Buffer (Cf. 3.5) ou les bases de données relationnelles. Le logiciel est ainsi capable de simuler des maquettes de sites paysagers de très grande dimension et dans un temps de calcul acceptable.



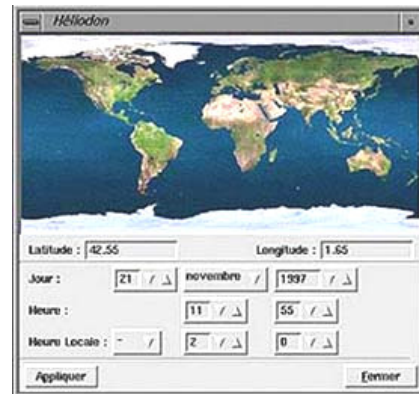
o Barre de menu et fonction annuler / rétablir

Une interface *mosaïque* donne accès aux principales fonctions du logiciel. Un historique complet de toutes les opérations effectuées est également conservé. Il est ainsi possible à tout moment d'annuler des opérations qui n'auraient pas eu l'effet escompté.

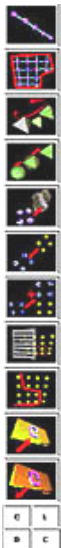


o Héliodon

Un héliodon simule la position exacte du soleil. Il permet de positionner automatiquement un soleil dans la scène et de simuler précisément les conditions d'éclairage correspondant à un jour et une heure donnée. Il est possible d'indiquer sur une carte fournie avec le logiciel, ou en spécifiant directement la longitude et la latitude, le lieu géographique du projet, la date et l'heure exacte de sa visualisation.



o Outils spéciaux pour les plantations



Des outils spécifiques permettent la plantation de grandes zones de végétation ou d'objets sous forme d'alignements, de plantations géométriques, et de bosquets sauvages. Il est possible d'y associer des espèces variées avec différentes densités de plantation.

o Création et gestion de groupes d'objets

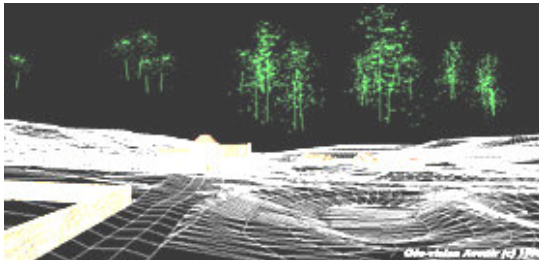
La fonction *groupe* permet d'associer sous une même identité des objets constituant la scène, quel que soit leur nature : arbres, architecture, terrains, ...

Ces groupes se gèrent ensuite comme des objets et facilite ainsi la constitution d'une grande base de données : forêts, haies, zone de végétation dense, groupe de bâtiments ou d'objets, alignements, zones particulières. Des outils de création et de gestion de ces groupes ont été spécifiquement développés pour la mise en scène de grandes maquettes de paysages.



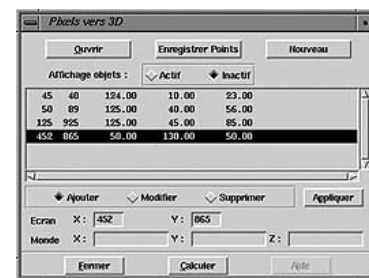
o Calcul automatique de l'altitude des objets

Une des difficultés de la manipulation d'objet en 3D est la vision correcte des distances et de la position des objets l'un par rapport aux autres. Notamment s'assurer qu'un arbre est bien planté au niveau du sol nécessite l'analyse de différents points de vue de la scène. C'est pourquoi une fonction de calcul automatique de l'altitude des objets a été développée dans AMAP Orchestra. Elle permet de planter automatiquement la totalité des objets ou des végétaux d'une scène sur le terrain sous-jacent.



o Rotation aléatoire

Afin d'augmenter le réalisme dans les grands ensembles d'objets de même nature (les forêts par exemple), une autre fonction permet d'attribuer une rotation aléatoire à des objets implantés de manière identiques dans une scène. Il est ainsi facile de donner de la diversité à un ensemble d'objets similaires.



o Calage d'une scène 3D sur une image en fond

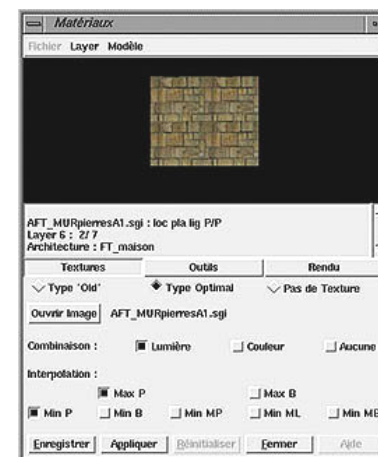


Dans des applications spécifiques, il est parfois nécessaire d'utiliser une image de fond (photo digitalisée ou image de synthèse) pour *habiller* l'arrière plan de la scène. AMAP Orchestra donne accès à un système de calage très performant. Après avoir choisi des points sur l'image de fond à l'aide de la souris, les coordonnées de l'image s'affichent dans la boîte de dialogue. Il suffit alors d'y associer des valeurs réelles relevées sur le site lors de la prise de vue ou sur un plan. Après calcul, le logiciel indique le calage exacte de la caméra, de son point de visé et de l'ouverture de la caméra.

o Plaquage de textures

Un module permet d'appliquer des textures (Cf. 3.4) sur les objets architecturaux et les différents éléments qui le constituent. Elles peuvent être modifiées en échelle, combinées à la couleur d'un objet, et réagir à la lumière ambiante de la scène.

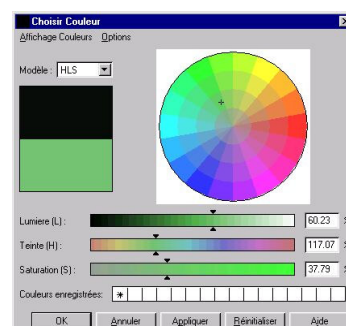
Les couleurs sont choisies sur un éditeur suivant différents modes chromatiques. Il donne également accès à des niveaux variés de teintes, de luminosités, et de contrastes. L'opérateur contrôle également l'ambient, le diffus, le spéculaire, l'émission et la brillance



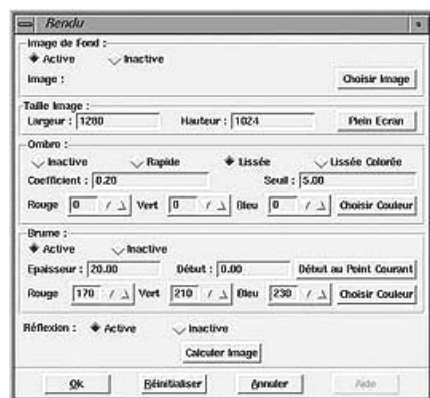
des objets qui composent la scène. Il est ainsi possible de simuler de l'eau ou une brillance en réglant son coefficient de réflexion et sa perturbation.

o Réglage des couleurs

Les couleurs des différents éléments constituant la scène peuvent être choisies à l'aide d'un outil fonctionnant en HLS, RVB, CMJ (Cf. 3.3). Il donne également accès à des niveaux variés de teinte, luminosité, et contraste. Le choix peut également se faire par nom, par tolérance ou par sélecteur chromatique.



o Paramétrisation des paramètres pour le rendu



Les images générées par AMAP Orchestra peuvent être calculées à partir d'une perspective unique, ou résultées d'un fichier animation décrivant le déplacement de l'observateur à l'intérieur de la scène virtuelle. Leur résolution peut atteindre 8000x8000 en 24 bits. Une boîte de dialogue regroupe les principaux modes de rendu.

La qualité des images peut être encore améliorée grâce à l'ajout d'effets spéciaux calculés par le logiciel comme les ombres portées, la brume atmosphérique, l'eau, la réflexion ... chaque effet peut faire l'objet de réglages précis.

5.2.3.3 Quelques réalisations de projets d'aménagement



Figure 100 Aménagement paysager d'un hôtel.



Figure 101 Projet de nouveau pont



Figure 102 Construction d'un rond-point.



Figure 103 Réaménagement d'une intersection : état actuel (à gauche), projet (à droite).



Figure 104 Paysages reconstitués : les Grandes Pyramides en Egypte (à gauche), le Taj Mahal en Inde (à droite).

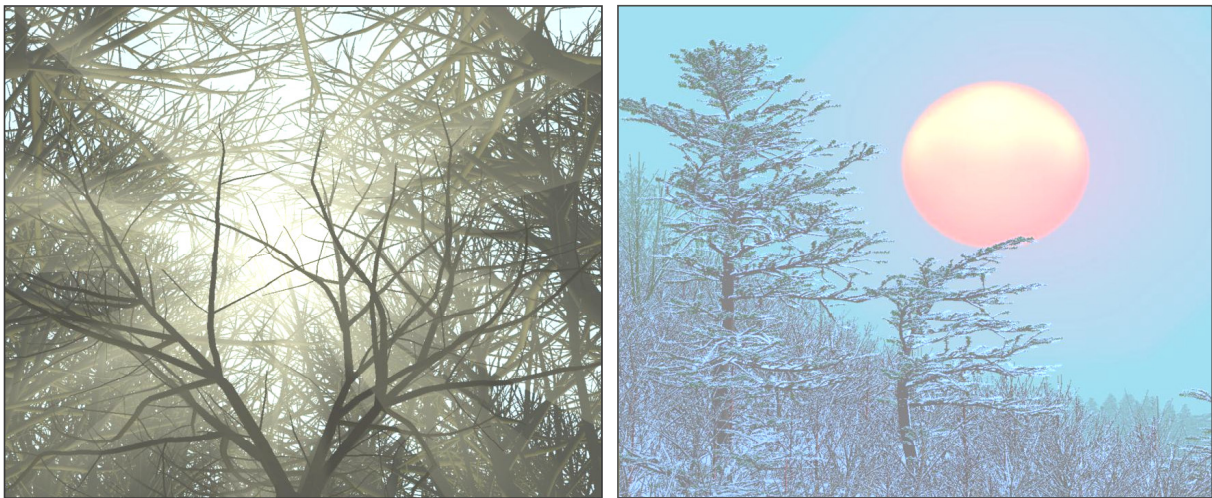


Figure 105 Paysages virtuels : lumière dans un sous-bois (à gauche), scène hivernale (à droite).

6 CONCLUSION

Après avoir introduit les notions de base de l'infographie, nous avons abordé le problème de la simulation et de la visualisation des plantes, en s'attachant, en particulier, aux modèles botaniques développés au laboratoire Amap.

Nous avons donc présenté succinctement quelques travaux développés dans ce cadre mais les activités du laboratoire Amap sont nombreuses et variées :

- Identifier les déterminants ontogénétiques, génétiques, phylogénétiques et environnementaux de la diversité et de la plasticité de la croissance et du fonctionnement des plantes. Evaluer la variabilité de l'architecture dans une perspective de sélection ou d'amélioration génétique.
Comprendre, modéliser et simuler le fonctionnement intégré des plantes et des peuplements végétaux afin de prédire leur production, en quantité et en qualité.
- Représenter les paysages et leur évolution dans une perspective d'aménagement et de gestion des espaces ruraux et urbains.
- Participer à la formation des étudiants et des chercheurs et à la diffusion des connaissances au travers de l'enseignement, de la formation, de l'édition et par le développement de logiciels de simulation de la croissance des arbres et des peuplements.

Un des objectifs de recherche est de comprendre et prédire le développement architectural des plantes et d'identifier le rôle respectif des facteurs génétiques, environnementaux et culturels au cours de leur croissance. A cette fin, les chercheurs s'attachent à poursuivre deux types d'objectifs :

- définir et mettre en œuvre les concepts et les méthodes nécessaires à la description, à la mesure, à l'analyse et à la compréhension de l'architecture des plantes ;
- évaluer les composantes de l'expression phénotypique et génétique de l'architecture ;
- analyser l'édification de l'architecture des plantes en fonction de processus physiologiques ;
- développer des outils de modélisation et de simulation intégrant des processus physiques, physiologiques et morphogénétiques pour prévoir le développement architectural des plantes dans des conditions agronomiques et climatiques variées.

Les recherches sur la composition des peuplements, la croissance et le développement des plantes fournissent des méthodes et des connaissances qu'il est ensuite possible d'assembler dans des outils quantitatifs de gestion qui opèrent au niveau des peuplements et des paysages. Ainsi, un autre objectif de recherche est :

- d'analyser et de modéliser l'organisation des couverts hétérogènes, en particulier les couverts forestiers et agroforestiers, tropicaux ou tempérés ;
- de décrire et de simuler le fonctionnement, la dynamique et la production de peuplements végétaux, forestiers, agricoles ou agroforestiers ;
- de simuler des paysages ruraux ou périurbains en élaborant des maquettes tridimensionnelles en vue :
 - de reconstituer et de visualiser l'évolution de l'environnement et les conséquences de stratégies de gestion des territoires,
 - et de servir de support à des recherches sur le fonctionnement des couverts végétaux à l'échelle du paysage.

Quels que soient les objectifs scientifiques, les problématiques de recherche abordées proviennent la plupart du temps de questions biologiques. Pour répondre à ces questions, il est parfois nécessaire de créer de nouveaux outils mathématiques et/ou informatiques. Dans certains cas, ces nouveaux outils sont eux-même porteurs d'une problématique scientifique à part entière. C'est pourquoi, une des activités d'AMAP a pour but de donner un panorama de cette activité de formalisation mathématique, de modélisation ou de production logicielle dans le laboratoire, transversalement à la description des objectifs scientifiques.

7 BIBLIOGRAPHIE

7.1 REFERENCES DANS LE TEXTE

- Amap, 2003.** *Amap : botanique et bioinformatique de l'Architecture des Plantes*. Web Site, <http://amap.cirad.fr>.
- Aono M. and Kunii T.L., 1984.** Botanical tree image generation. IEEE Computer Graphics and Applications, 4(5).
- Aries D. and Viennot X.G., 1990.** Simulation of Organic Structures, In Eurographics'90, Montreux, Suisse.
- Blaise, F., 1991.** *Simulation du parallélisme dans la croissance des plantes et application*. Thèse de 3ième cycle, Université Louis Pasteur, Strasbourg.
- Boole, G., 1854.** «An investigation into the laws of thought, on which are founded the mathematical theories of logic and probabilities ».
- Bosanac B., 1990.** *TREE : A knowledge-based system for generating images of trees*, Thèse d'université, Harvard.
- Barnsley M.F., 1988.** « Fractals Everywhere ». Academic Press, New-York.
- Bionatics®, 2000.** *Bionatics : Born from the AMAP research*. Web Site, <http://www.bionatics.com>.
- Blinn J.F., Newel M.E., 1976.** Texture and Reflection in Computer Generated Images. CACM, 19(10), 542-547.
- Bloomenthal J., 1985.** Modeling the Mighty Maple. Computer Graphics, 19(3), 305-311.
- Bresenham J.E., 1965.** « Algorithm for computer control of a digital plotter. » IBM System Journal, 4 : 25-30.
- Bui-Tong P., 1975.** Illumination for Computer Generated Pictures. CACM, 18(6), 311-317.
- Catmull E. 1974.** A Subdivision Algorithm for Computer Display of Curved Surfaces. Ph.D. Thesis, Report UTEC-CSc-4133, Computer Science Department, University of Utah, Salt Lake City, UT.
- D'Arcy Thompson, 1966.** « On Growth and Form ». Cambridge at the University Press.
- Devillers O., 1988.** Méthodes d'optimisation du tracé de rayons. Thèse de 3ième cycle, Université de Paris-Sud, Centre d'Orsay.
- Dulk (den) J.A. 1989.** *The Interpretation of Remote Sensing, a feasibility study*. Master's thesis, Wageningen Agricultural University, Neederland.
- Eyrolles G., 1986.** *Synthèse d'images figuratives d'arbres par des méthodes combinatoires*. Thèse de 3ième cycle, Université de Bordeaux I, Bordeaux.
- Eyrolles G., Viennot G. and Françon J., 1986.** « Combinatoire pour la synthèse d'images réalistes de plantes. » 2ième semaine de l'Image Electronique, Nice.
- Fisher J.B. and Honda H., 1979a.** Branch geometry and effective leaf area: a study of terminalia branching pattern - 1 - theoretical trees, Amer. J. Bot., 66, 633-644.
- Fisher J.B. and Honda H., 1979b.** Branch geometry and effective leaf area: a study of terminalia branching pattern - 2 - survey of real trees, Amer. J. Bot., 66, 645-655.
- Foley J.D., Dam (van) A., Feiner S.K., Hughes J.F., 1990.** «Computer Graphics : Principles and practice». Addison Wesley, 1174p.
- Fourcaud T., Lac P., 1996.** Mechanical analysis of the form and internal stresses of a growing tree by the finite element method. PD-Vol. 77, ESDA Proceedings, Vol 5 Bioengineering, Engin A.E. Ed., ASME : 213-220.
- Fournier A., 1987.** Modeling of Natural Phenomena, Siggraph'87, Course, Anaheim.
- Fournier A., Fussel D., Carpenter L., 1982.** Computer Rendering of Stochastic Models. CACM, 25(6), June 1982, 371-384.
- Fournier A., Reeves W.T., 1986.** A simple model of ocean waves. Siggraph 86, 75-84.
- Françon J., 1984.** Arbres et nombres de Strahler dans diverses sciences, Revue du Palais de la Découverte, 12(120), 29-36.
- Françon J., 1990.** Sur la modélisation informatique de l'architecture et du développement des végétaux, 2nd Colloque International « L'Arbre », Institut de Botanique, Montpellier.
- Frijters D. and Lindenmayer A., 1974.** A model for the growth and flowering of *Aster novaeangliae* on the basis of table (1,0) L-systems, In: L_Systems, G. Rozenberg and A. Salomaa, Springer-Verlag, 24-52.
- Galimberti R., Montanari U., 1969.** An Algorithm, for Hidden Line Elimination. CACM, 12 (4), 206-211.
- Gardner G.Y., 1985.** Visual simulation of clouds. Siggraph 85, 297-303.
- Goral C.M., Torrance K.E., Greenberg D.P., Battaille B., 1984.** Modeling the Interaction of Light Between Diffuse Surfaces. Siggraph 84, 213-222.
- Gouraud H., 1971.** Continuous shading of curved surfaces. IEEE Trans. on Computers, C-20 (6), 623-629.

- Green N., 1989.** Voxel space automata: modeling with stochastic growth processes in voxel space. *Computer Graphics*, 23 (3), 175-184.
- Greenworks, 1996.** *Greenworks organic software.*, WEB site, <http://www.greenworks.de>.
- Guillier, F., 1998.** *Histoire de l'Informatique.* Web Site, <http://www.histoire-informatique.org>.
- Hanan J.S., 1995.** Virtual plants – integrating architectural and physiological models. In *Proceedings of Int. Congr. Modelling and Simulation*, Australia.
- Hogeweg P. and Hesper B., 1974.** A model study on biomorphological description, *Pattern Recognition*, 6, 165-179.
- Honda H., 1971.** Description of the form of trees by the parameters of the tree-like body: effects of the branching angle and the branch length on the shape of the tree-like body, *J. Theor. Biol.*, 31, 331-338.
- Horton R.E., 1945.** Erosional Development of Streams and their Drainage Basins: Hydrophysical Approach to Quantitative Morphology, *Bull. of the Geological Soc. of America*, 56, 275-370.
- Jaeger M., 1987.** *Représentation et simulation de croissance des végétaux.* Thèse de 3^{ème} cycle, Université Louis Pasteur, Strasbourg.
- Kawaguchi Y., 1982.** A Morphological Study of the Forme of Nature, *Computer Graphics*, 16(3).
- Kozłowski T.T., 1949.** Light and water in relation to growth and competition of forest tree species, *Ecol. Monogr.*, 19, 207-231.
- Leibnitz, W.G., 1697.** « Origine radicale des choses ».
- Leroudier J., 1980.** « La simulation à événements discrets », Monographies d'informatique de l'AFCEI, Editions Hommes et Techniques.
- Lienhardt P., 1987.** *Modélisation et évolution de surfaces libres.* Thèse de 3^{ème} cycle Université Louis Pasteur, Strasbourg.
- Lienhardt P. and Françon J., 1987.** Synthèse d'images de feuilles végétales, *In 3^{ème} Colloque Image*, Paris.
- Lindenmayer A., 1968a.** Mathematical model for cellular interaction in developments - I : filaments with one side inputs, *J. Theor. Biol.*, 18, 280-299.
- Lindenmayer A., 1968b.** Mathematical model for cellular interaction in developments - II: simple and branching filaments with two-side inputs, *J. Theor. Biol.*, 18, 300-315.
- Linterman B., Deussen O., 1998.** Interactive modeling of plants, *IEEE Computer Graphics and Applications*, vol., n°, 1998.
- Magenat-Thalmann N., Thalmann D., 1985.** *Computer animation : Theory and Practice.* Springer-Verlag, Tokyo.
- Mandelbrot B., 1977.** « Fractals, form, chance and dimension ». W.H. Freeman and Cie, San Francisco.
- Mandelbrot B., 1982.** « The fractal geometry of nature ». W.H. Freeman and Cie, San Francisco.
- Mech R., Prusinkiewicz P., 1996.** Visual Models of Plants : Interacting with Their Environment. *Computer Graphics (SIGGRAPH '96 Proceedings)*, 28:397-410.
- Onyx computing, 1995.** *Tree professional*, WEB site, <http://www.OnyxTree.com/professionalw.html>.
- Oppenheimer E., 1986.** Real Time Design and Animation of Fractal Plants and Trees, *Computer graphics*, 20(4), 55-64.
- Persi D. and Shahshahani M., 1984.** « Products of Random Matrices and Computer Image Generation », Stanford University Preprint, Science.
- Prusinkiewicz P. 1993.** Modeling and visualization of biological structures. In *Proceedings of Graphics Interface '93*, pages 128-137.
- Prusinkiewicz P., Hammel M., Mech R. 1994.** Visual Models of Morphogenesis : A Guide Tour. *World Wide Web server*, <http://www.cpsc.ucalgary.ca/Redirect/bmv/vmm-deluxe/TitlePage.html>.
- Prusinkiewicz P., Lindenmayer A., 1990.** « The algorithmic beauty of plants ». Springer-Verlag, New York. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.
- Queau Ph., 1986.** « Eloge de la simulation. » *Collection milieux* CHAMP VALLON, Institut National de l'Audiovisuel.
- Reffye (de) P., Edelin C., Françon J., Jaeger M., Puech C., 1988.** Plant models faithful to botanical structure and development, *Computer Graphics*, 22 (4).
- Reeves W.T., 1983.** Particle Systems: A technique for modeling a class of fuzzy objects. *Computer Graphics*, 2, 91-108.
- Reeves W.T. and Blau R., 1985.** Approximate and Probabilistic Algorithm for Shading and Rendering Structured Partial Systems. *Computer Graphics*, 19, 313-322.
- Robertson B., 1989.** Technology in bloom, *Computer Graphics World*, 73-81.
- Room P.M., Hanan J.S., 1994.** Virtual cotton : A new tool for research, management and training. In *Proceedings of World Cotton Res. Conf. I*, Brisbane.
- Room P.M., Hanan J.S., Prusinkiewicz P., 1996.** Virtual plants : new perspectives for ecologists, pathologists and agricultural scientists. *Trends in plant science*, Elsevier Science, 1 :33-38.
- Shannon, 1948.** « Théorie mathématique de la communication ».
- Sillion F., 1995.** A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters, *IEEE Transactions on Visualization and Computer Graphics*, 3 (1).

- Smith A.R., 1984.** Plants, Fractals and Formal Languages. Siggraph 84, 1-10.
- Stevens P.S., 1978.** « Les formes dans la nature », Seuil.
- Strahler A.N., 1952.** Hypsometric (Area-Altitude) Analysis of Erosional Topology, Bull. of the Geological Soc. of America, 63, 1117-1142.
- Viennot X.G., 1990.** « Mélanges. Livre dédié à M.-P. Schützenberger ». In Trees, Hermès, Paris.
- Warnock J., 1969.** A Hidden-Surface Algorithm for Computer Generated Half-Tone Pictures. Technical Report TR 4-15, NTIS AD-753 671, Computer Science Department, University of Utah, Salt Lake City, UT.
- Watkins G.S., 1970.** A Real Time Visible Surface Algorithm. Ph.D. Thesis, Technical Report UTEC-CSc-70-101, NTIS AD-762 004, Computer Science Department, University of Utah, Salt Lake City, UT.
- Whitted T., 1980.** An Improved Illumination Model for Shaded Display. CACM, 23(6), 343-349.

7.2 REFERENCES COMPLEMENTAIRES

- Barthélémy D., Blaise F., Fourcaud T., Nicolini E., 1995.** « Modélisation et simulation de l'architecture des arbres : bilan et perspectives. » *Rev. For. Fr.*, XLVII, N° hors Série : 71-96.
- Blaise F., Barczy J.F., Jaeger M., Dinouard P., Reffye P. de, 1998.** Simulation of the growth of plants - Modeling of metamorphosis and spatial interactions in the architecture and development of plants. In: Cyberworlds, Kunii T.L., Luciani A. Eds., Tokyo (JAP), Springer : 81-109
- Blaise F., Houllier F., Reffye de P. 1996.** « Simulation of tree architecture and growth in a forest stand: AMAPpara software. » In G. Nepveu (Ed.) : IUFRO WP S5.01.04 Workshop, *Connection between silviculture and wood quality through modelling approaches and simulation softwares* (Hook, Sweden, 13-17/06/94), INRA, Nancy: 46-55.
- Blaise F., Reffye de P., 1994.** « AMAP et la croissance secondaire. » In : *Architecture, Structure, Mécanique de l'Arbre*, 7ème Séminaire Interne, Montpellier (FRA), 05-07/04/94, LMGC Equipe Bois; pp. 111-122
- Blaise F., Reffye de P., 1994.** « Simulation de la croissance des arbres et influence du milieu : le logiciel AMAPpara. » In : Actes du 2ème Colloque Africain sur la Recherche en Informatique, Ouagadougou (Burkina Faso), 12-18 oct. 1994, Tankoano J (ed.), ORSTOM Editions, Collection Colloques et Séminaires, Paris: 61-75.
- Blaise F., Reffye de P., Houllier F. 1997.** « Sawing logs in virtual trees. » In : G. Nepveu (Ed.) : IUFRO WP S5.01.04 Second Workshop, *Connection between silviculture and wood quality through modelling approaches and simulation softwares* (Berg-en-Dal, South Africa, 26-30/08/96): 203-212 [communication et texte].
- Bouchon J., Houllier F., 1995.** « Une brève histoire de la modélisation de la production des peuplements forestiers : place des méthodes architecturales. » In : J. Bouchon (Ed.) : *Architecture des arbres fruitiers et forestiers* (23-25/11/93, Montpellier), INRA, Les Colloques, 74 : 17-2.
- Bouchon J., Reffye de P., Barthélémy D., 1997.** « Modélisation et simulation de l'architecture des végétaux. » Sciences Update, INRA Editeur, 435p.
- Fourcaud T., 1995.** « Analyse du comportement mécanique d'une plante en croissance par la méthode des éléments finis. » thèse de doctorat de l'Université Bordeaux I, n°1194, 148p.
- Fourcaud T., Castera P., Lac P., 1994.** « Stress analysis of a growing plant with a finite element method » In : *Workshop IUFRO*, Hook (Sweden), June 1994.
- Fourcaud T., Lac P., 1995.** « Mechanical analysis of the form and internal stresses of a growing tree by the finite element method. » In : *Architecture, Structure, Mécanique de l'Arbre*, 8ème Séminaire Interne, Montpellier (FRA), 12-13/07/95, LMGC Equipe Bois; pp. 65-73
- Fourcaud, T., Lac, P., 1997.** Analyse mécanique d'une branche d'arbre durant sa croissance. In : 13ème Congrès français de mécanique, Futuroscope, Poitiers (FRA), 01-05/09/97 : 37-40
- Jaeger M., Leban J. M., Chemouny S., Saint André L., 1999** « 3D stem reconstruction from CT scan exams ». Soumis et accepté au *Third Workshop IUFRO WP S5.01-04. Biological improvement of wood properties*, La Londe-Les-Maures, France, 5-12 septembre 1999.
- Lecoustre R., Saïto K., Kumagay Y., Honjo T., Reffye de P, Ishida Y., 1997.** « Simulation d'un paysage forestier à partir d'un SIG et de la technique de modélisation des plantes. » In: *Modélisation et Simulation de l'Architecture des Végétaux*, Bouchon J., de Reffye P., Barthélémy D. Eds.; Paris (FRA) : Science Update, INRA Editions : 425-434.
- Prusinkiewicz P. 1994.** Visual Models of Morphogenesis. In *Artificial Life*, 1(1/2):67-74.
- Prusinkiewicz P., James M., Mech R. 1994a.** Synthetic Topiary. *Computer Graphics (SIGGRAPH '94 Proceedings)*, 28:351-358.
- Prusinkiewicz P., Remphrey W., Davidson C., Hammel M. 1994b.** Modeling the architecture of expanding FRAXINUS PENNSYLVANICA shoots using L-systems. *Canadian Journal of Botany*, 72:701-714.
- Reffye de P., Houllier F., Auclair D., Barczy J.-F., Dauzat J. 1997.** « Tree architecture as a link between

- processes and forest growth. » *IUFRO Workshop Empirical and process-based models for forest tree and stand growth simulation* (Oeiras, Portugal, 21-27/09/97), 13 pp.
- Reffye de P., Houllier F., Blaise F. 1997.** « Modelling plant growth and architecture: some recent advances and applications to agronomy and forestry. » In: L.F.M. Marcelis : *Second International Symposium on Models for Plant Growth, Environmental Control and Farm Management in Protected Cultivation* (Wageningen, The Netherlands, 25-28/08/97), ISHS, 12 pp. (texte intégral)
- Reffye de P., Houllier F., Blaise F. 1998.** « Modelling plant growth and architecture: some recent advances and applications to agronomy and forestry. » *Acta Horticulturae*, **456**: 105-116.
- Reffye de P., Houllier F., Blaise F., Barthélémy D., Dauzat J., 1997.** « Modélisation et simulation de la croissance d'une architecture végétale : approche morphogénétique expérimentale. » In : *Tendances nouvelles en modélisation pour l'environnement*, Journées du Programme Environnement, Vie et Sociétés du CNRS, Blasco F. (Ed.), Elsevier : 91-112 (texte intégral).
- Reffye de P., Houllier F., Blaise F., Barthélémy D., Dauzat J., Auclair D., 1995.** « A model simulating above- and below- ground tree architecture with agroforestry applications. » *Agroforestry Systems*, 30 : 175-197.
- Reffye P. de, Houllier F., Blaise F., Barthélémy D., Dauzat J., Auclair D., 1997.** « Modélisation et simulation de la croissance d'une architecture végétale : une approche morphogénétique expérimentale. » In : Blasco F. (ed.). *Tendances nouvelles en modélisation pour l'environnement*, Vie et Sociétés du CNRS, Elsevier, Paris : 91-112.
- Reffye P. de, Houllier F., Blaise F., Fourcaud T., 1997.** « Essai sur les relations entre l'architecture d'un arbre et la grosseur de ses axes végétatifs. » In : *Modélisation et Simulation de l'Architecture des Végétaux*, Bouchon J., de Reffye P., Barthélémy D. Eds.; Paris (FRA) : Science Update, INRA Editions : 255-423
- Reffye de P., Houllier F., Blaise F., Fourcaud T., 1997.** « Principe de la simplification de l'architecture des arbres à l'échelle d'un peuplement forestier hétérogène. » séminaire *GIP ECOFOR Forêts hétérogènes*, 4-6 juin 1997, Seignosse, France.